



Trabajo Fin de Grado

Modelado y simulación de cuadricópteros Modeling and simulation of quadrotors

Autor

Miguel Ángel Fustero Carreras

Director

Enrique Teruel Doñate

Escuela de Ingeniería y Arquitectura
2019

Lista de Símbolos

$\{g\} = \{X_B, X_Y, X_Z\}$	Base de coordenadas fija al suelo	
$\{o\} = \{X, X, X\}$	Base de coordenadas solidara al quadricóptero	
$X = [x, y, z]^T$	Vector posición en la base $\{g\}$	
x	Posición en el Eje X_B	[m]
y	Posición en el Eje Y_B	[m]
z	Posición en el Eje Z_B	[m]
$\dot{X} = [\dot{x}, \dot{y}, \dot{z}]^T$	Vector velocidad lineal en la base $\{g\}$	
\dot{x}	Velocidad lineal en el Eje X_B	[m/s]
\dot{y}	Velocidad lineal en el Eje Y_B	[m/s]
\dot{z}	Velocidad lineal en el Eje Z_B	[m/s]
$\ddot{X} = [\ddot{x}, \ddot{y}, \ddot{z}]^T$	Vector aceleración lineal en la base $\{g\}$	
\ddot{x}	Aceleración lineal en el Eje X_B	[m/s ²]
\ddot{y}	Aceleración lineal en el Eje Y_B	[m/s ²]
\ddot{z}	Aceleración lineal en el Eje Z_B	[m/s ²]
$\Psi = [\phi, \theta, \psi]^T$	Vector orientación en la base $\{g\}$	
ϕ	Orientación en el Eje X_B	[rad]
θ	Orientación en el Eje Y_B	[rad]
ψ	Orientación en el Eje Z_B	[rad]
$\dot{\Psi} = [\dot{\phi}, \dot{\theta}, \dot{\psi}]^T$	Vector velocidad angular en la base $\{g\}$	
$\dot{\phi}$	Velocidad angular en el Eje X_B	[rad/s]
$\dot{\theta}$	Velocidad angular en el Eje Y_B	[rad/s]
$\dot{\psi}$	Velocidad angular en el Eje Z_B	[rad/s]
$\ddot{\Psi} = [\ddot{\phi}, \ddot{\theta}, \ddot{\psi}]^T$	Vector aceleración angular en la base $\{g\}$	
$\ddot{\phi}$	Aceleración angular en el Eje X_B	[rad/s ²]
$\ddot{\theta}$	Aceleración angular en el Eje Y_B	[rad/s ²]
$\ddot{\psi}$	Aceleración angular en el Eje Z_B	[rad/s ²]
$v = [v_x, v_y, v_z]^T$	Vector velocidad lineal en la base $\{o\}$	
v_x	Velocidad lineal en el Eje X	[m/s]
v_y	Velocidad lineal en el Eje Y	[m/s]
v_z	Velocidad lineal en el Eje Z	[m/s]

$\dot{v} = [\dot{v}_x, \dot{v}_y, \dot{v}_z]^T$	Vector aceleración lineal en la base $\{o\}$	
\dot{v}_x	Aceleración lineal en el Eje X	$[\text{m/s}^2]$
\dot{v}_y	Aceleración lineal en el Eje Y	$[\text{m/s}^2]$
\dot{v}_z	Aceleración lineal en el Eje Z	$[\text{m/s}^2]$
$R = [R_x, R_y, R_z]^T$	Vector velocidad angular en la base $\{o\}$	
R_x	Velocidad angular en el Eje X	$[\text{rad/s}]$
R_y	Velocidad angular en el Eje Y	$[\text{rad/s}]$
R_z	Velocidad angular en el Eje Z	$[\text{rad/s}]$
$\dot{R} = [\dot{R}_x, \dot{R}_y, \dot{R}_z]^T$	Vector aceleración angular en la base $\{o\}$	
\dot{R}_x	Velocidad angular en el Eje X	$[\text{rad/s}^2]$
\dot{R}_y	Velocidad angular en el Eje Y	$[\text{rad/s}^2]$
\dot{R}_z	Velocidad angular en el Eje Z	$[\text{rad/s}^2]$
J	Tensor de inercia del cuadricóptero	
I_{xx}	Momento de inercia en torno al Eje X	$[\text{kg} \cdot \text{m}^2]$
I_{yy}	Momento de inercia en torno al Eje Y	$[\text{kg} \cdot \text{m}^2]$
I_{zz}	Momento de inercia en torno al Eje Z	$[\text{kg} \cdot \text{m}^2]$
$\tau = [\tau_x, \tau_y, \tau_z]^T$	Vector de pares	
τ_x	Par en torno al Eje X	$[\text{N} \cdot \text{m}]$
τ_y	Par en torno al Eje Y	$[\text{N} \cdot \text{m}]$
τ_z	Par en torno al Eje Z	$[\text{N} \cdot \text{m}]$
T	Empuje	$[\text{N}]$
g	Gravedad	$[\text{m/s}^2]$
$\Omega = [\Omega_1, \Omega_2, \Omega_3, \Omega_4]^T$	Vector velocidad de giro de los rotores	
$\Omega_s = [\Omega_1^2, \Omega_2^2, \Omega_3^2, \Omega_4^2]^T$	Vector velocidad de giro de los rotores al cuadrado	
Ω_1	Velocidad de giro del rotor 1	$[\text{rad/s}]$
Ω_2	Velocidad de giro del rotor 2	$[\text{rad/s}]$
Ω_3	Velocidad de giro del rotor 3	$[\text{rad/s}]$
Ω_4	Velocidad de giro del rotor 4	$[\text{rad/s}]$
$\dot{\Omega}_1$	Derivada velocidad de giro del rotor 1	$[\text{rad/s}^2]$
$\dot{\Omega}_2$	Derivada velocidad de giro del rotor 2	$[\text{rad/s}^2]$
$\dot{\Omega}_3$	Derivada velocidad de giro del rotor 3	$[\text{rad/s}^2]$
$\dot{\Omega}_4$	Derivada velocidad de giro del rotor 4	$[\text{rad/s}^2]$
W_g	Término giroscópico	$[\text{rad/s}]$
b	Constante de empuje de los rotores	$[\text{N} \cdot \text{s}^2 / \text{rad}^2]$
F_1	Fuerza del rotor 1	$[\text{N}]$
F_2	Fuerza del rotor 2	$[\text{N}]$
F_3	Fuerza del rotor 3	$[\text{N}]$
F_4	Fuerza del rotor 4	$[\text{N}]$
d	Constante de arrastre de los rotores	$[\text{N} \cdot \text{m} \cdot \text{s}^2 / \text{rad}^2]$
M_1	Momento inducido por el rotor 1	$[\text{N} \cdot \text{m}]$
M_2	Momento inducido por el rotor 2	$[\text{N} \cdot \text{m}]$
M_3	Momento inducido por el rotor 3	$[\text{N} \cdot \text{m}]$
M_4	Momento inducido por el rotor 4	$[\text{N} \cdot \text{m}]$

m_0	Masa total del cuadricóptero	[kg]
m	Masa de cada uno de los rotores	[kg]
M	Masa del cuerpo del cuadricóptero	[kg]
r	Radio del cuerpo del cuadricóptero	[m]
l_p	Diámetro del rotor	[m]
l	Longitud del brazo del cuadricóptero	[m]
v_1	Tensión de entrada del motor 1	[V]
v_2	Tensión de entrada del motor 2	[V]
v_3	Tensión de entrada del motor 3	[V]
v_4	Tensión de entrada del motor 4	[V]
K_m	Constante mecánica del motor	[N·m/A]
K_e	Constante eléctrica del motor	[V·m/rad]
R	Resistencia	[Ω]
F_R	Fuerza de rozamiento traslacional	[kg·m/s ²]
A_R	Fuerza de rozamiento rotacional	[N·s/m]
A_T	Matriz coeficientes rozamiento traslacional	[N·s/m]
A_R	Matriz coeficientes rozamiento rotacional	[N·s/m]
a_T	Coefficiente de fricción traslacional	[N·s/m]
a_R	Coefficiente de fricción rotacional	[N·s/m]
L	Inductancia de la bobina	[H]
i	Corriente	[A]
T_m	Par generador por el motor	[kg · m]
T_l	Par resistente de la carga	[kg · m]

Índice de figuras

2.2. Fuerzas y pares actuando sobre el sistema. ^[4]	6
2.3. Velocidad lineal (v) y angular (R) sobre la base $\{o\}$. ^[4]	6
2.4. Geometría del cuadricóptero. ^[4]	9
2.5. Esquema motor DC brushless. ^[3]	12
3.1. Movimientos del cuadricóptero. ^[1]	18
3.3. Desplazamiento en el eje Z.	18
3.5. Rotación en el eje Z.	19
3.6. Rotación en el eje Z con el efecto del rozamiento M_R .	21
3.7. Rotación en el eje Z con el efecto del rozamiento M_R .	22
3.7. Rotación en el eje Z con freno.	22
3.8. Traslación Eje X, rotación Eje Y.	23

3.10. Traslación Eje Y, rotación Eje X.	24
4.1. Esquema de control de los motores.	26
4.2. Controlador PI con anti Wind Up.	26
4.3. Control PD, simulación posición eje Z.	28
4.4. Control PD, simulación ángulos ψ , θ y ψ	29
4.5. Control PID, simulación posición eje Z.	30
4.6. Control PID, simulación ángulos ψ , θ y ψ	30
4.7. Control PID, simulación modelo detallado.	31
4.9. Control de posición, referencia en x.	34
4.10. Control de posición, referencia en y.	34
4.11. Control de posición, referencia en z.	34
4.12. Control de posición, referencia en ψ	35
4.13. Control de posición sobre el modelo detallado.	35
4.16. Control en espacio de estados, sistema linealizado.	41
4.15. Control en espacio de estados, sistema detallado.	41
4.17. Comparativa ángulo ψ	42

Índice general

Lista de Símbolos	I
Índice de figuras	VI
1. Introducción	1
2. Modelado físico	3
2.1. Cambios de base	4
2.2. Modelado del movimiento lineal	5
2.3. Modelado del movimiento angular	7
2.4. Fuerzas y momentos actuantes	8
2.5. Dinámica del motor	11
2.6. Implementación en Modelica	13

3. Verificación del modelo físico	17
3.1. Traslación en el eje Z	18
3.2. Rotación sobre el eje Z	19
3.3. Traslación en el eje X	22
3.4. Traslación en el eje Y	23
4. Diseño del control	25
4.1. Control de los motores	26
4.2. Control PD y PID	27
4.3. Control de posición	31
4.4. Linealización modelo físico	35
4.5. Control en el Espacio de Estados	39
5. Conclusiones	43
Bibliografía	45

Capítulo 1

Introducción

El objetivo del presente trabajo es la realización del modelado de un cuadricóptero y su simulación en Modelica, un software de código libre orientado a la simulación de sistemas complejos cuyo uso es muy interesante, ya que permite la conexión de todo tipo de subcomponentes. Para ello, en primer lugar y antes de comenzar con el modelado físico, se realiza una primera fase de aprendizaje con el software a través de una serie de ejercicios propuestos por el tutor de este trabajo, Enrique Teruel. Estos ejercicios sirven de iniciación para comenzar a manejar el programa y familiarizarse con el uso del mismo.

Una vez finalizada esta primera fase de aprendizaje comienza la etapa de documentación, en la que se investiga sobre el tema en cuestión, el modelado de cuadricópteros. Se comienza recabando información sobre los cuadricópteros para comprender mejor su funcionamiento y los componentes de los que constan. Durante este periodo se estudia documentación facilitada por el tutor de estudios previos que tratan sobre una temática similar, para comenzar a orientar el proyecto.

El siguiente paso es comenzar con el modelado físico. Esta parte del trabajo consiste en modelar como se comporta un cuadricóptero, básicamente se trata de lograr un modelo matemático que simule el comportamiento físico de un cuadricóptero. El objetivo es lograr una serie de ecuaciones que relacionen las entradas del sistema, es decir, los cuatro motores con los movimientos que realiza el cuadricóptero en el espacio.

Una vez se tiene diseñado el modelo lo mas ajustado posible a la realidad, se van a diseñar diferentes controladores para lograr que el cuadricóptero se comporte como

se desee. Por ejemplo, conseguir que se desplace a un punto concreto del espacio o que gire un determinado ángulo en una dirección.

Se van a diseñar y simular varios controladores diferentes para compararlos y ver las diferencias existentes entre ellos y si hay alguno que sea más interesante que los demás y en qué situaciones lo es.

Por último, se obtendrá mediante linealización en torno al punto de trabajo habitual un modelo físico simplificado, mucho más sencillo que el propuesto inicialmente, a partir del cual se diseñará un controlador en el Espacio de Estados. Este control es diseñado a partir del modelo simplificado, y se simulará también sobre el modelo más realista para comprobar si es posible utilizar el modelo sencillo para diseñar el control e implementarlo en cuadricópteros reales. En caso de que la respuesta del sistema fuese buena, se buscarán las limitaciones que tiene el control, en caso de que las tuviera.

Capítulo 2

Modelado físico

A lo largo del presente capítulo, se va a presentar y explicar el modelo físico desarrollado para implementar y simular en Modelica durante todo el proyecto. En primer lugar, y antes de comenzar con el desarrollo del modelo físico, es necesario definir las dos bases en las que se va a trabajar durante todo el proyecto. Todas las magnitudes físicas que se empleen a lo largo de este documento van a estar siempre referidas a una de las dos bases.

Ambas están formadas por tres ejes cartesianos. La primera, fija al suelo, se denotará de ahora en adelante $\{g\} = \{X_B, Y_B, Z_B\}$ y la segunda, móvil, situada en el centro de masas del cuadricóptero y que se mueve de manera solidaria a este, se denominará $\{o\} = \{X, Y, Z\}$.

Para posicionar la base $\{o\}$ respecto de $\{g\}$, se define el vector $x = (x, y, z)^T$ que indica la posición de la base móvil respecto de la fija. Una vez posicionada la base, se definen los tres ángulos de Tait-Bryan necesarios para orientar la base móvil $\{o\}$ respecto de la base fija $\{g\}$. Estos tres ángulos son ϕ , que hace referencia al giro en el eje X, el ángulo θ , que indica el giro en el eje Y y el ángulo ψ que es el giro en el eje Z. Estos tres ángulos se definen de tal manera que son positivos siguiendo el criterio de la regla de la mano derecha. Nótese que estos ángulos también son habitualmente denominados como $\phi \equiv \text{roll}$, $\theta \equiv \text{pitch}$ y $\psi \equiv \text{yaw}$.

2.1 Cambios de base

Al trabajar en dos bases diferentes, es fundamental definir las matrices de cambio de base para poder pasar de una a otra de manera ágil y rápida. Primero se construye la matriz de cambio con respecto a cada uno de los ejes, es decir, $R(X, \phi)$, $R(Y, \theta)$ y $R(Z, \psi)$.

$$R(X, \phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix} \quad (2.1)$$

$$R(Y, \theta) = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \quad (2.2)$$

$$R(Z, \psi) = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.3)$$

Una vez obtenidas las tres matrices de cambio elementales, construimos la matriz de cambio global $R(\phi, \theta, \psi)$. Esta matriz se obtiene multiplicando las tres matrices que se acaban de obtener, tal y como se detalla en la ecuación 2.4.

$$R(\phi, \theta, \psi) = R(X, \phi)R(Y, \theta)R(Z, \psi) = \begin{bmatrix} \cos \psi \cos \theta & \cos \psi \sin \theta \sin \phi - \sin \psi \cos \phi & \cos \psi \sin \theta \cos \phi + \sin \psi \sin \phi \\ \sin \phi \cos \theta & \sin \psi \sin \theta \sin \phi + \cos \psi \cos \phi & \sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi \\ -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi \end{bmatrix} \quad (2.4)$$

Esta matriz de cambio permite cambiar la velocidad lineal de la base móvil a la base fija como se indica en la ecuación 2.5.

$$\dot{x} = R(\phi, \theta, \psi)v \quad (2.5)$$

Se procede ahora a calcular la matriz de cambio necesaria para cambiar de base la velocidad angular. En esta ocasión para realizar el cambio usaremos la matriz

$R_A^{-1} = (\phi, \theta, \psi)$. A continuación la ecuación 2.6^[10] indica como calcular dicha matriz, y en ecuación 2.7 se muestra la relación existente entre la velocidad angular en base móvil y base fija.

$$R_A^{-1}(\phi, \theta, \psi) = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi / \cos \theta & \cos \phi / \cos \theta \end{bmatrix} \quad (2.6)$$

$$\dot{\Psi} = R_A^{-1}(\phi, \theta, \psi) R \quad (2.7)$$

2.2 Modelado del movimiento lineal

Una vez definidos y explicados los dos sistemas de coordenadas y las matrices de cambio, el siguiente paso es la definición de las ecuaciones que describen la dinámica del cuadricóptero. Se comienza definiendo el modelo dinámico para el movimiento lineal. La ecuación 2.8^[9] plantea de forma general la contribución de las distintas fuerzas que se ejercen sobre el sistema en la dinámica del mismo.

$$\dot{v} = g \begin{bmatrix} \sin \theta \\ -\sin \phi \cos \theta \\ -\cos \phi \cos \theta \end{bmatrix} - Sv + \begin{bmatrix} 0 \\ 0 \\ T/m_0 \end{bmatrix} \quad (2.8)$$

En la ecuación 2.8, el vector $v = (v_x, v_y, v_z)^T$ expresa la velocidad del cuadricóptero en la base $\{o\}$ y el vector $\dot{v} = (\dot{v}_x, \dot{v}_y, \dot{v}_z)^T$ la aceleración del mismo en dicha base. T es el empuje que siempre va en la dirección positiva del eje Z en la base $\{o\}$ y g la aceleración de la gravedad, que lleva la dirección negativa del eje Z_B en la base $\{g\}$. Siendo el vector $R = (R_x, R_y, R_z)^T$ la velocidad angular en la base $\{o\}$, la matriz S se define según se indica en la ecuación 2.9^[9]. En la Figura 2.2 se indican las fuerzas T y g y los pares τ_x, τ_y y τ_z que actúan sobre el sistema. En la figura 2.3 se muestra como se define la velocidad lineal (v) y angular (R) en la base $\{o\}$.

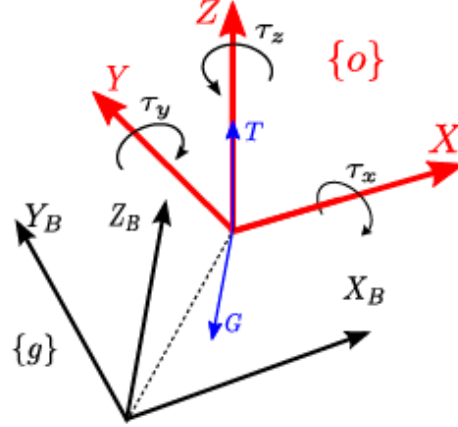


Figura 2.2: Fuerzas y pares actuando sobre el sistema.^[4]

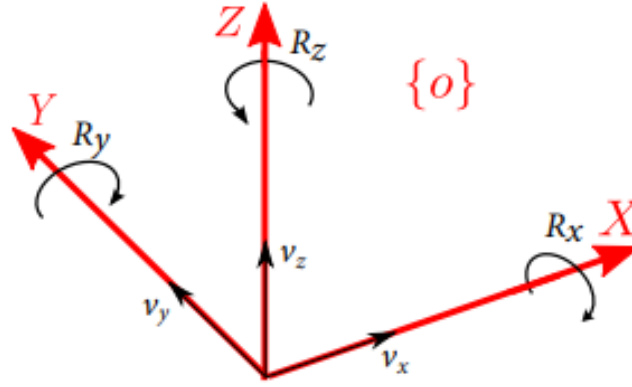


Figura 2.3: Velocidad lineal (v) y angular (R) sobre la base $\{o\}$.^[4]

$$S = \begin{bmatrix} 0 & -R_z & R_y \\ R_z & 0 & -R_x \\ -R_y & R_x & 0 \end{bmatrix} \quad (2.9)$$

Sustituyendo en la ecuación 2.8, se obtiene:

$$\begin{bmatrix} \dot{v}_x \\ \dot{v}_y \\ \dot{v}_z \end{bmatrix} = g \begin{bmatrix} \sin \phi \\ -\sin \phi \cos \theta \\ -\cos \phi \cos \theta \end{bmatrix} - \begin{bmatrix} 0 & -R_z & R_y \\ R_z & 0 & -R_x \\ -R_y & R_x & 0 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ T/m_0 \end{bmatrix} \quad (2.10)$$

Y desarrollando la expresión, quedan las siguientes 3 ecuaciones:

$$\begin{cases} \dot{v}_x = g \sin \theta + v_y R_z - v_z R_y \\ \dot{v}_y = -g \sin \phi \cos \theta + v_z R_x - v_x R_z \\ \dot{v}_z = -g \cos \phi \cos \theta + v_x R_y - v_y R_x + T/m_0 \end{cases} \quad (2.11)$$

Las tres ecuaciones obtenidas en la expresión 2.11 son las que se emplean para definir los movimientos de traslación del cuadricóptero.

2.3 Modelado del movimiento angular

Una vez obtenidas las ecuaciones que definen el modelo dinámico del movimiento lineal, se van a definir ahora las ecuaciones que modelan la dinámica rotacional del cuadricóptero. Se parte de la ecuación 2.12^[4], en la que la matriz J es la formada por los momentos de inercia. En este caso, y tal y como se ha definido la posición de los ejes (ver Figura 2.4), tiene la forma que se especifica en la ecuación 2.13 y J^{-1} su inversa. El vector $\tau = (\tau_x, \tau_y, \tau_z)$ representa cada uno de los pares en los ejes de la base $\{o\}$, I_{zzm} es el momento de inercia del rotor y W_g el termino giroscópico que se define como se muestra en la ecuación 2.14.

$$\dot{R} = J^{-1} \left(\tau - SJR - S \begin{bmatrix} 0 \\ 0 \\ I_{zzm} W_g \end{bmatrix} \right) \quad (2.12)$$

$$J = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \quad (2.13)$$

$$W_g = -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4 \quad (2.14)$$

Siendo I_{xx} , I_{yy} y I_{zz} los momentos de inercia respecto a los ejes X, Y y Z del sistema local $\{o\}$ de coordenadas. Para el cálculo de los momentos de inercia, se utilizan las ecuaciones^[4] 2.15 y 2.16.

$$I_{xx} = I_{yy} = \frac{2Mr^2}{5} + 2ml^2 \quad (2.15)$$

$$I_{zz} = \frac{2Mr^2}{5} + 4ml^2 \quad (2.16)$$

Por su parte, I_{zzm} se puede aproximar por medio de la ecuación 2.17^[9].

$$I_{zzm} = \frac{m_p l_p^2}{12} \quad (2.17)$$

Sustituyendo en la ecuación 2.12 se obtiene la ecuación 2.18 y desarrollando, se deducen las 3 ecuaciones que se indican a continuación (expresión 2.19) y que rigen el comportamiento dinámico del movimiento rotacional del cuadricóptero.

$$\begin{aligned} \begin{bmatrix} \dot{R}_x \\ \dot{R}_y \\ \dot{R}_z \end{bmatrix} = \begin{bmatrix} \frac{1}{I_{xx}} & 0 & 0 \\ 0 & \frac{1}{I_{yy}} & 0 \\ 0 & 0 & \frac{1}{I_{zz}} \end{bmatrix} \left(\begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} - \begin{bmatrix} 0 & -R_z & R_y \\ R_z & 0 & -R_x \\ -R_y & R_x & 0 \end{bmatrix} \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} R_x \\ R_y \\ R_z \end{bmatrix} - \right. \\ \left. - \begin{bmatrix} 0 & -R_z & R_y \\ R_z & 0 & -R_x \\ -R_y & R_x & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ I_{zzm} W_g \end{bmatrix} \right) \end{aligned} \quad (2.18)$$

$$\begin{cases} I_{xx} \dot{R}_x = R_z R_y (I_{zz} - I_{yy}) + \tau_x - R_y I_{zzm} \Omega \\ I_{yy} \dot{R}_y = R_z R_x (I_{xx} - I_{zz}) + \tau_y + R_x I_{zzm} \Omega \\ I_{zz} \dot{R}_z = R_y R_x (I_{yy} - I_{xx}) + \tau_z \end{cases} \quad (2.19)$$

2.4 Fuerzas y momentos actuantes

Una vez definidas las ecuaciones que rigen el comportamiento dinámico del cuadricóptero, se procede a explicar cómo se calculan las fuerzas y momentos generados por los motores en el cuadricóptero. Estas fuerzas y momentos son claves, ya que modificando los valores de estas a través de las entradas de nuestro sistema, que son

las tensiones de los cuatro motores, se puede lograr que actúen sobre el cuadricóptero las fuerzas y momentos necesarios para que este se mueva como deseamos. En la Figura 2.4 se muestra la numeración y el sentido de giro de los motores del modelo que se va a simular además de la posición de los ejes de la base local de coordenadas.

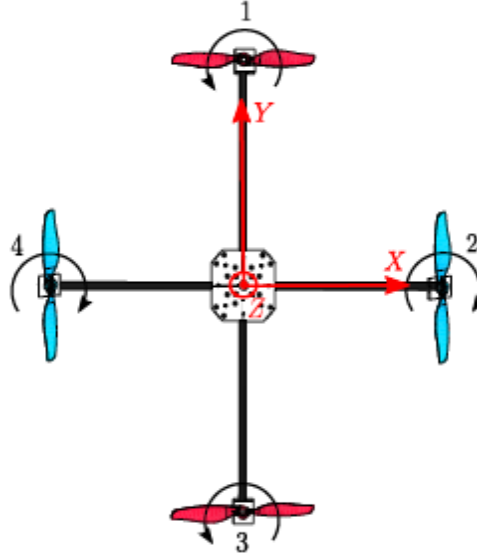


Figura 2.4: Geometría del cuadricóptero.^[4]

A continuación se calcula la fuerza ejercida por cada uno de los motores como se indica en la ecuación 2.20^[7], siendo $b = 9,9865 \cdot 10^{-6} N \cdot s^2 / rad^2$ la constante de empuje. Estas fuerzas F_i siempre van en la dirección positiva del eje Z .

$$F_i = b\Omega_i^2 \quad (2.20)$$

$$\begin{cases} F_1 = b\Omega_1^2 \\ F_2 = b\Omega_2^2 \\ F_3 = b\Omega_3^2 \\ F_4 = b\Omega_4^2 \end{cases} \quad (2.21)$$

Por otra parte, cada uno de los motores produce un momento inducido que se calcula según la ecuación 2.22, siendo $d = 1,5978 \cdot 10^{-7} N \cdot m \cdot s^2 / rad^2$ la constante de arrastre de cada uno de los rotores.

$$M_i = d\Omega_i^2 \quad (2.22)$$

$$\begin{cases} M_1 = d\Omega_1^2 \\ M_2 = d\Omega_2^2 \\ M_3 = d\Omega_3^2 \\ M_4 = d\Omega_4^2 \end{cases} \quad (2.23)$$

En la expresión 2.24^[2] se muestran las expresiones utilizadas para calcular el empuje T y los pares en cada uno de los ejes, τ_x , τ_y y τ_z en función de las fuerzas y momentos obtenidos en las ecuaciones 2.21 y 2.23.

$$\begin{cases} \tau_x = l(F_1 - F_3) \\ \tau_y = l(F_4 - F_2) \\ \tau_z = -M_1 + M_2 - M_3 + M_4 \\ T = F_1 + F_2 + F_3 + F_4 \end{cases} \quad (2.24)$$

Los pares en los ejes X e Y se consiguen haciendo girar los motores situados sobre el eje contrario a diferentes velocidades, esta diferencia de velocidad provoca una diferencia de fuerzas de empuje que a su vez es la que provoca que se genere el par. Este par generado es igual a la diferencia de fuerzas multiplicada por la distancia de aplicación de esta fuerza, que en este caso es la longitud del brazo, l . El empuje se calcula realizando el sumatorio de las fuerzas provocadas por cada uno de los cuatro motores. Por último, el par en el eje Z , al estar definido como positivo según la regla de la mano derecha, se calcula sumando los momentos inducidos que van en la misma dirección (motores 3 y 4) y restando los que van en dirección contraria (motores 1 y 2). En la figura 2.4 se muestra la dirección de giro de los motores, lo que provoca que el momento inducido tenga la dirección contraria.

Se procede ahora a construir la matriz M , que relaciona el vector de velocidades angulares de los rotores al cuadrado, $\Omega_s = [\Omega_1^2, \Omega_2^2, \Omega_3^2, \Omega_4^2]^T$ con los pares en cada uno de los ejes y el empuje. Con esta matriz, conocidas las velocidades de giro de cada uno de los rotores, se obtienen directamente las fuerzas y pares que actúan sobre el cuadricóptero, como se muestra en la ecuación 2.25.

$$\begin{bmatrix} T \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = M\Omega_s \quad (2.25)$$

Con las ecuaciones explicadas anteriormente podemos construir la matriz M , quedando como se muestra en la ecuación 2.26.

$$M = \begin{bmatrix} b & b & b & b \\ bl & 0 & -bl & 0 \\ 0 & -bl & 0 & bl \\ -d & d & -d & d \end{bmatrix} \quad (2.26)$$

Teniendo b y d que tienen los valores anteriormente comentados, y la longitud del brazo, que para el cuadricóptero que se simula tiene un valor de $l = 0,211 \text{ m}$, la matriz M queda como se muestra en la expresión 2.27.

$$M = \begin{bmatrix} 9,9865 \cdot 10^{-6} & 9,9865 \cdot 10^{-6} & 9,9865 \cdot 10^{-6} & 9,9865 \cdot 10^{-6} \\ 2,1072 \cdot 10^{-6} & 0 & -2,1072 \cdot 10^{-6} & 0 \\ 0 & -2,1072 \cdot 10^{-6} & 0 & 2,1072 \cdot 10^{-6} \\ -1,5978 \cdot 10^{-7} & 1,5978 \cdot 10^{-7} & -1,5978 \cdot 10^{-7} & 1,5978 \cdot 10^{-7} \end{bmatrix} \quad (2.27)$$

2.5 Dinámica del motor

A lo largo de esta sección se va a proceder a modelar los motores que lleva montados el cuadricóptero que se va a simular. Lo habitual en cuadricópteros es que se empleen motores eléctricos de corriente continua sin escobillas, también llamados motores DC brushless. Estos motores son ideales para su aplicación en drones, ya que proporcionan un buen par con muy poca fricción. El modo de modelar la dinámica para un motor DC brushless es el mismo que si se tratase de un motor de corriente continua convencional.

Para comenzar con el modelado del motor, se muestra en la Figura 2.5 un esquema del circuito eléctrico y mecánico equivalente.

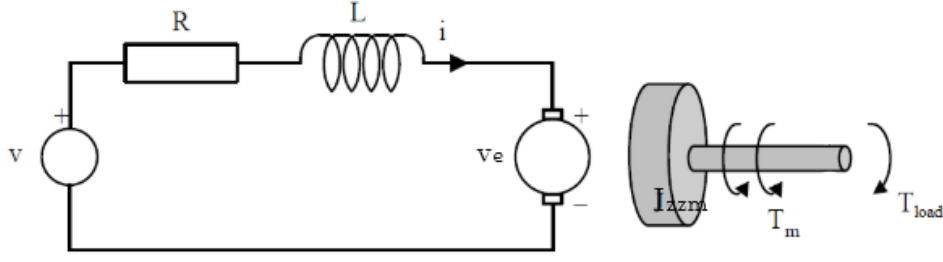


Figura 2.5: Esquema motor DC brushless.^[3]

Aplicando la ley de Kirchhoff se obtiene la ecuación 2.28. En la que R es la resistencia del motor, L la inductancia, i la corriente, v la tensión aplicada y el término $K_m \cdot \Omega$ hace referencia a la fuerza contraelectromotriz generada.

$$v = R \cdot i + L \cdot \frac{di}{dt} + K_m \cdot \Omega \quad (2.28)$$

El valor de la inductancia suele ser relativamente pequeño, y su efecto se suele despreciar^[8], quedando la expresión anterior como se muestra en la ecuación 2.29.

$$v = R \cdot i + K_m \cdot \Omega \quad (2.29)$$

Despejando el valor de la corriente i se obtiene la siguiente expresión que después será de utilidad.

$$i = \frac{v - K_m \Omega}{R} \quad (2.30)$$

El planteamiento de la ecuación mecánica que modela el motor según la Figura 2.5 es el mostrado en la ecuación 2.31, en la que T_m es el par producido por el motor y es igual a la constante mecánica del motor multiplicada por la corriente $K_e i$ y T_l es el par resistente provocado por la carga y que se puede aproximar con la expresión $d\Omega^2$, que es la misma que la ecuación 2.22.

$$I_{zzm}\dot{\Omega} = T_m - T_l \quad (2.31)$$

$$I_{zzm}\dot{\Omega} = K_e i - M \quad (2.32)$$

Sustituyendo el valor de i obtenido en la ecuación 2.30 e introduciendolo en la ecuación 2.32 y generalizando para cada uno de los cuatro motores, se obtiene la ecuación 2.33.

$$I_{zzm}\dot{\Omega}_i + \frac{K_m K_e}{R} \Omega_i = \frac{K_m}{R} v_i - M_i \quad (2.33)$$

Particularizando para cada uno de los cuatro motores, se obtienen las cuatro ecuaciones que a continuación se muestran.

$$\begin{cases} I_{zzm}\dot{\Omega}_1 + \frac{K_m K_e}{R} \Omega_1 = \frac{K_m}{R} v_1 - M_1 \\ I_{zzm}\dot{\Omega}_2 + \frac{K_m K_e}{R} \Omega_2 = \frac{K_m}{R} v_2 - M_2 \\ I_{zzm}\dot{\Omega}_3 + \frac{K_m K_e}{R} \Omega_3 = \frac{K_m}{R} v_3 - M_3 \\ I_{zzm}\dot{\Omega}_4 + \frac{K_m K_e}{R} \Omega_4 = \frac{K_m}{R} v_4 - M_4 \end{cases} \quad (2.34)$$

2.6 Implementación en Modelica

Una vez se ha presentado y explicado todo el modelo físico que se va a utilizar en el presente trabajo, el siguiente paso es introducir el modelo en el software en el que se van a hacer las simulaciones, Modelica.

Para ello, y antes de comenzar, es necesario dar valores a todas las constantes que se han numerado a lo largo del capítulo. Estos valores se muestran en la tabla 2.1.

Variable	Valor	Unidades
I_{xx}	0.0128	$\text{kg} \cdot \text{m}^2$
I_{yy}	0.0128	$\text{kg} \cdot \text{m}^2$
I_{zz}	0.0239	$\text{kg} \cdot \text{m}^2$
I_{zzm}	$4.3011 \cdot 10^{-5}$	$\text{kg} \cdot \text{m}^2$
R	0.1107	Ω
K_m	0.01	$\text{N} \cdot \text{m} / \text{A}$
K_e	0.01	$\text{V} \cdot \text{s} / \text{rad}$
m_0	1.32	kg
m	0.125	kg
M	0.82	kg
r	0.07	m
l_p	0.065	m
g	9.80665	m / s^2

Cuadro 2.1: Valores de las variables para la simulación.^[4]

Para simplificar la implementación en Modelica, se va a dividir el modelo físico en varios bloques, conectados entre sí a través de sus entradas y salidas. De este modo, el funcionamiento del sistema es exactamente el mismo que si se agregasen todas las ecuaciones en el mismo bloque, pero se simplifica notablemente la programación del bloque y además, se facilita la detección de posibles errores.

El bloque final obtenido, que incluye todas las ecuaciones desarrolladas a lo largo del capítulo tiene como entradas el vector de tensiones, $V = [v_1, v_2, v_3, v_4]^T$ formado por las tensiones a las que se alimentan los cuatro motores y tiene como salidas los vectores posición, $X = [x, y, z]^T$, orientación, $\Psi = [\phi, \theta, \psi]^T$ y sus respectivas derivadas $\dot{X} = [\dot{x}, \dot{y}, \dot{z}]^T$ y $\dot{\Psi} = [\dot{\phi}, \dot{\theta}, \dot{\psi}]^T$. Además, también se define como salida la velocidad de giro de los motores, $\Omega = [\Omega_1, \Omega_2, \Omega_3, \Omega_4]^T$. Este bloque se llama MAV, son las siglas de Multirrotor Aerial Vehicle.

El primer bloque que se diseña, incluye las ecuaciones que modelan el movimiento del cuadricóptero. Posee como entradas el vector de pares $\tau = [\tau_x, \tau_y, \tau_z]^T$ y el empuje T . A este bloque se le conecta otro que incluye las ecuaciones de las que se deducen los valores de los pares y el empuje y posee como entrada el vector de velocidades de giro de los motores $\Omega = [\Omega_1, \Omega_2, \Omega_3, \Omega_4]^T$. Por último, se añade un último bloque en el que se incluye la dinámica de los motores (idénticos), que tiene como entradas el vector tensiones $V = [v_1, v_2, v_3, v_4]^T$ y como salidas el vector velocidad de giro de los motores.

Una vez se ha introducido todo el modelo en Modelica, se da paso al siguiente capítulo, en el que se detalla el proceso que se va a seguir para llevar a cabo la

verificación del modelo, es decir, que el modelo es válido, se comporta como se espera y no hay ningún error en la programación del mismo en el software.

Capítulo 3

Verificación del modelo físico

Para verificar el modelo descrito en el capítulo anterior se van a realizar diferentes simulaciones con valores concretos para las entradas, las tensiones de los cuatro rotores, de tal manera que deberían de provocar que el cuadricóptero reaccionase de un determinado modo. Se van a ensayar individualmente los giros en torno a los tres ejes y los desplazamientos en los mismos. Si de todas estas pruebas se obtienen los resultados esperados, se puede concluir que el modelo físico es correcto y está bien implementado en Modelica.

Todos los ensayos que se hacen a lo largo de este capítulo son con todas las variables inicializadas a cero excepto la velocidad de giro de los motores, que comienzan girando a $\Omega = 569 \text{ rad/s}$, velocidad para la que el empuje generado por los rotores, se iguala a la acción de la gravedad. Esto supone que el cuadricóptero se mantiene en su posición inicial y ambos sistemas de coordenadas, el sistema fijo al suelo $\{g\}$ y el solidario al cuadricóptero $\{o\}$ son coincidentes. Antes de comenzar con las simulaciones, en la Figura 3.1 se muestra cómo deben ser accionados los motores del cuadricóptero para que haga los movimientos que se desean (desplazamientos y rotaciones en cada uno de los tres ejes). Resumiendo lo que en la imagen se explica, para realizar movimientos en el eje Z, hay que accionar los cuatro motores a la misma velocidad, si se quiere ascender, esta deberá ser superior a 569 rad/s para ascender, e inferior para descender. Para hacer rotaciones sobre el eje Z, se giran a menor velocidad dos de los motores. Por último, para desplazarse en X e Y, se accionan 2 motores a la misma velocidad, uno a una velocidad superior y el otro a una velocidad inferior, de manera que el cuadricóptero se inclina y avanza en la dirección del motor que gira más lento.

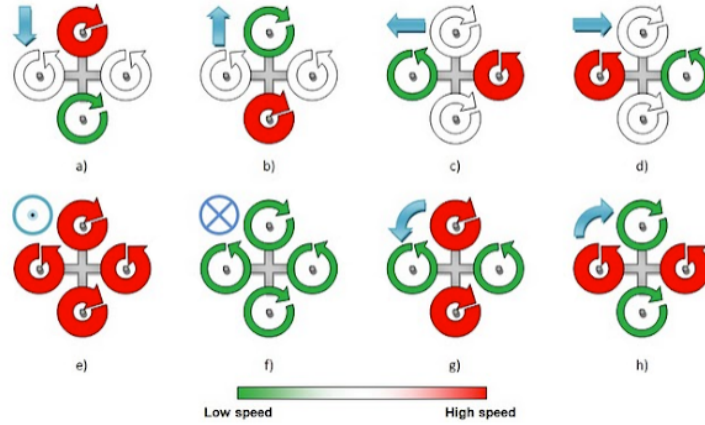


Figura 3.1: Movimientos del cuadricóptero.^[1]

3.1 Traslación en el eje Z

En primer lugar se accionan los cuatro motores a la vez con una tensión de 6.27 V, que supone una velocidad de giro $\Omega = 569,26 \text{ rad/s}$. Para esta velocidad de giro de los motores, y encontrándose el dron en posición inicial de reposo, la fuerza de empuje debería igualarse a la fuerza de la gravedad y el cuadricóptero mantenerse en el aire, sin ascender ni descender en el eje Z, como se comprueba que sucede.

A continuación, en la Figuras 3.3 se muestra como asciende el cuadricóptero en el eje X cuando se alimentan los motores a 6.5 V.

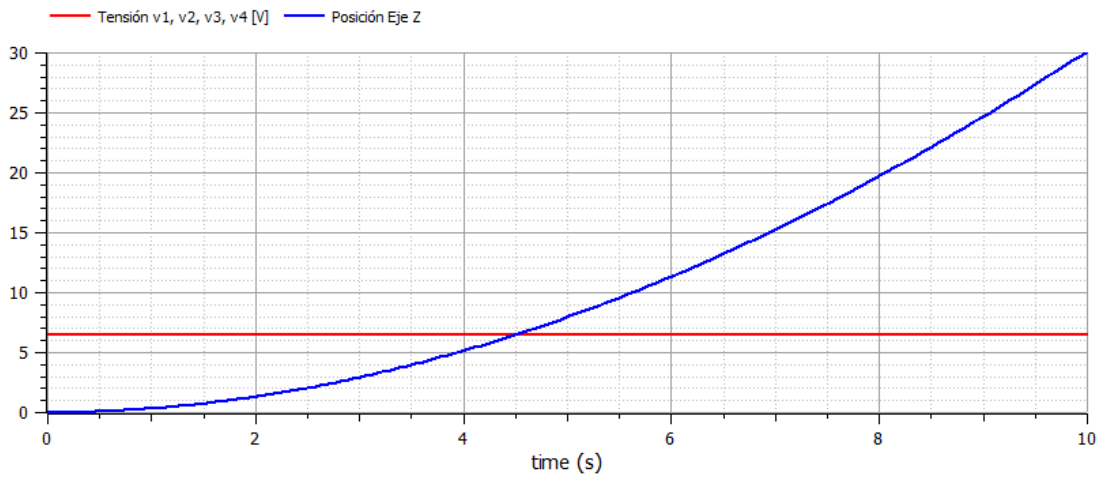


Figura 3.3: Desplazamiento en el eje Z.

3.2 Rotación sobre el eje Z

En la siguiente simulación que se va a acometer se pretende verificar que el cuadricóptero gira sobre su propio eje Z cuando se accionan en cruz dos motores a una velocidad y otros dos a otra.

Los motores 1 y 3 se accionan durante los primeros 5 segundos a 5.98 V y los motores 2 y 4 a 6.54 V, estas tensiones son elegidas porque su promedio es 6.27 y de esta manera, el cuadricóptero sigue estacionario en Z. Pasados los 5 segundos, los cuatro motores se accionan a 6.27 V. Esta diferencia de tensión entre los motores durante los primeros 5 segundos, genera un par en el eje Z, τ_z , que provoca que el cuadricóptero gire.

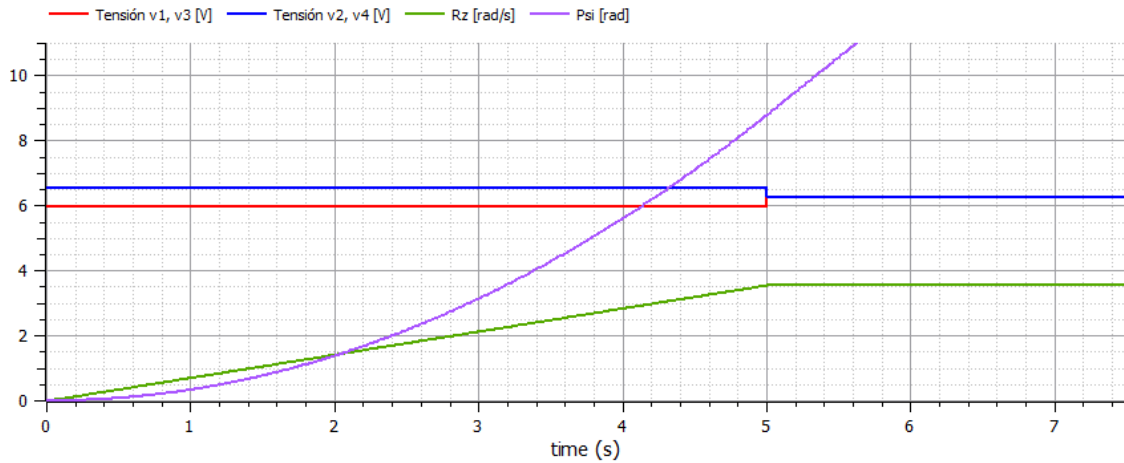


Figura 3.5: Rotación en el eje Z.

Analizando el resultado obtenido que se muestra en la Figura 3.5, se observa que una vez pasados los 5 segundos, el cuadricóptero sigue girando puesto que se observa que el giro en el eje Z aumenta de forma lineal. Esto significa que una vez ha desaparecido el par τ_z , el cuadricóptero sigue girando. Esto se debe a que en el modelo físico planteado anteriormente no se ha introducido el efecto del rozamiento del aire en la rotación y traslación del dron.

Por lo tanto, se va a proceder a modificar los modelos dinámicos propuestos durante el capítulo 2 y se van a tener en cuenta los efectos de rozamiento con el aire, y se analizará hasta que punto es importante esta fuerza en el movimiento del cuadricóptero. Las ecuaciones que se van a modificar son las ecuaciones 2.8 y 2.12,

que describen los movimientos de traslación y rotación, a las que se les va a añadir el efecto de fricción con el aire.

Asumiendo que las velocidades a las que se va a desplazar el dron son bajas, se puede aproximar la fuerza de fricción del aire como una constante por la velocidad lineal en cada uno de los tres ejes del sistema de referencia solidario al dron. En la ecuación 3.1 se muestra la expresión con la que se calcula esta fuerza F_R . Siendo $a_T = 0,04 \text{ N} \cdot \text{s}/\text{m}$ la constante de fricción traslacional.^[6]

$$F_R = -A_T v = - \begin{bmatrix} a_T & 0 & 0 \\ 0 & a_T & 0 \\ 0 & 0 & a_T \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} \quad (3.1)$$

Añadiendo esta fuerza a la ecuación del modelo dinámico de traslación calculada en el capítulo anterior, se obtiene la ecuación 3.2 y desarrollando las tres ecuaciones siguientes.

$$\dot{v} = g \begin{bmatrix} \sin \theta \\ -\sin \phi \cos \theta \\ -\cos \phi \cos \theta \end{bmatrix} - S v + \begin{bmatrix} 0 \\ 0 \\ \frac{T}{m_0} \end{bmatrix} + \frac{1}{m_0} F_R \quad (3.2)$$

$$\begin{cases} \dot{v}_x = g \sin \theta + v_y R_z - v_z R_y - \frac{a_T v_x}{m_0} \\ \dot{v}_y = -g \sin \phi \cos \theta + v_x R_z - v_z R_x - \frac{a_T v_y}{m_0} \\ \dot{v}_z = -g \cos \phi \cos \theta + v_x R_y - v_y R_x + \frac{T}{m_0} - \frac{a_T v_z}{m_0} \end{cases} \quad (3.3)$$

El cálculo del efecto del rozamiento del aire a la rotación del cuadricóptero se realiza por medio de la ecuación 3.4, que tiene la misma forma que la fuerza de rozamiento traslacional. Siendo $a_R = 6,4 \cdot 10^{-4} \text{ N} \cdot \text{s}/\text{m}$ la constante de fricción rotacional.^[6]

$$M_R = -A_R R = \begin{bmatrix} a_R & 0 & 0 \\ 0 & a_R & 0 \\ 0 & 0 & a_R \end{bmatrix} \begin{bmatrix} R_x \\ R_y \\ R_z \end{bmatrix} \quad (3.4)$$

Agregando esta contribución a la ecuación del modelo dinámico de rotación calculada en el capítulo anterior, se obtiene la ecuación 3.5 y desarrollando, las tres ecuaciones que se muestran seguidamente.

$$\dot{R} = J^{-1} \left(\tau - SJR - S \begin{bmatrix} 0 \\ 0 \\ I_{zzm}W_g \end{bmatrix} \right) + \frac{1}{m_0} M_R \quad (3.5)$$

$$\begin{cases} I_{xx}\dot{R}_x = R_z R_y (I_{zz} - I_{yy}) + \tau_x - R_y I_{zzm} \Omega - \frac{a_R R_x}{m_0} \\ I_{yy}\dot{R}_y = R_z R_x (I_{xx} - I_{zz}) + \tau_y + R_x I_{zzm} \Omega - \frac{a_R R_y}{m_0} \\ I_{zz}\dot{R}_z = R_y R_x (I_{yy} - I_{xx}) + \tau_z - \frac{a_R R_z}{m_0} \end{cases} \quad (3.6)$$

Una vez agregadas estas contribuciones al modelo en Modelica, se procede a repetir la simulación para comprobar la rotación sobre el eje Z, cuyo resultado se muestra en la Figura 3.6. Se comprueba que el rozamiento con el aire prácticamente no influye en la rotación del cuadricóptero. Esto se debe a que la contribución del momento M_R en la ecuación 3.5 es muy pequeño.

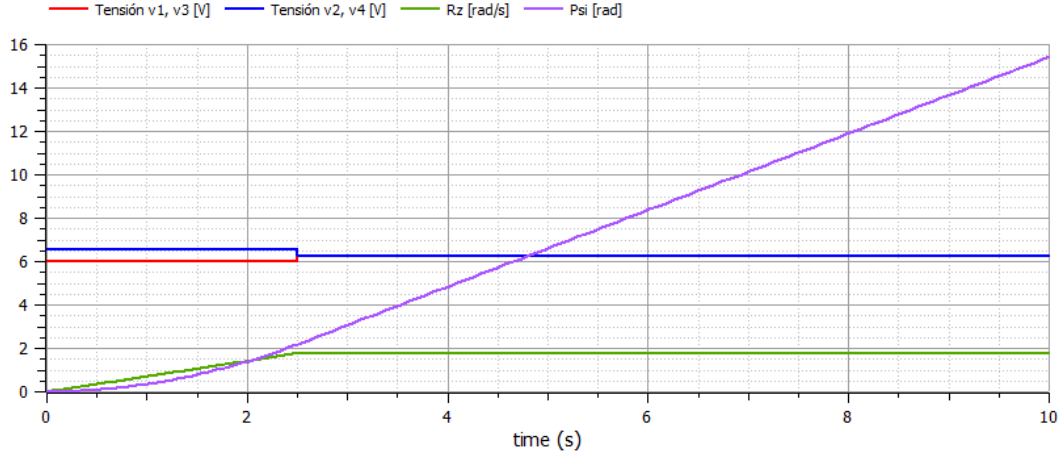


Figura 3.6: Rotación en el eje Z con el efecto del rozamiento M_R .

A pesar de que el momento M_R es muy pequeño, si se ha modelado correctamente, en algún momento debería apreciarse que la velocidad se ve reducida. Para ello, en la Figura 3.7 se simulan 200 s. Como era de esperar se aprecia que la velocidad se reduce ligeramente.

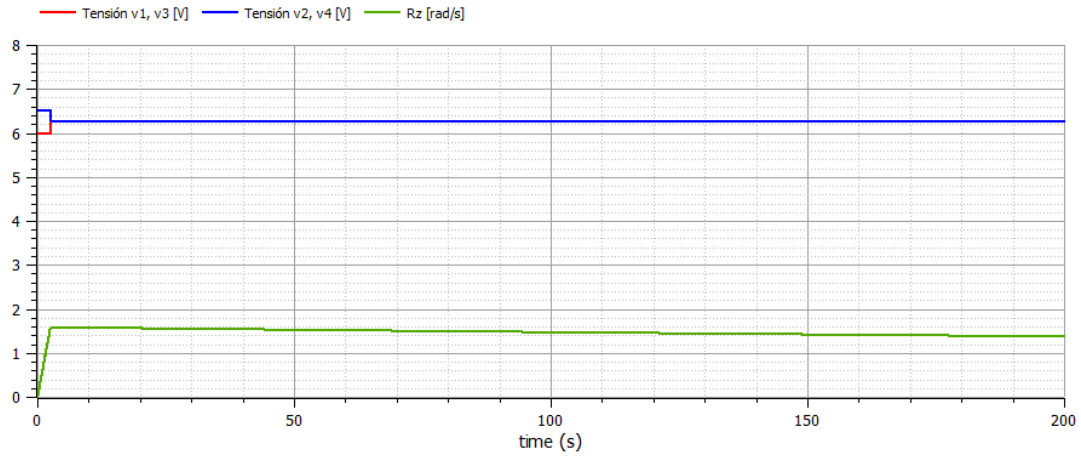


Figura 3.7: Rotación en el eje Z con el efecto del rozamiento M_R .

Por lo tanto, para frenar el movimiento iniciado en un tiempo razonable, será necesario aplicar un par negativo en torno a ese eje. Esto se muestra en la Figura 3.7, en la que tras llevar un tiempo girando el cuadricóptero a velocidad constante, se alimentan los motores para generar el par inverso, lo que logra reducir a cero la velocidad de rotación en torno al eje Z, y el ángulo de giro ψ permanece constante.

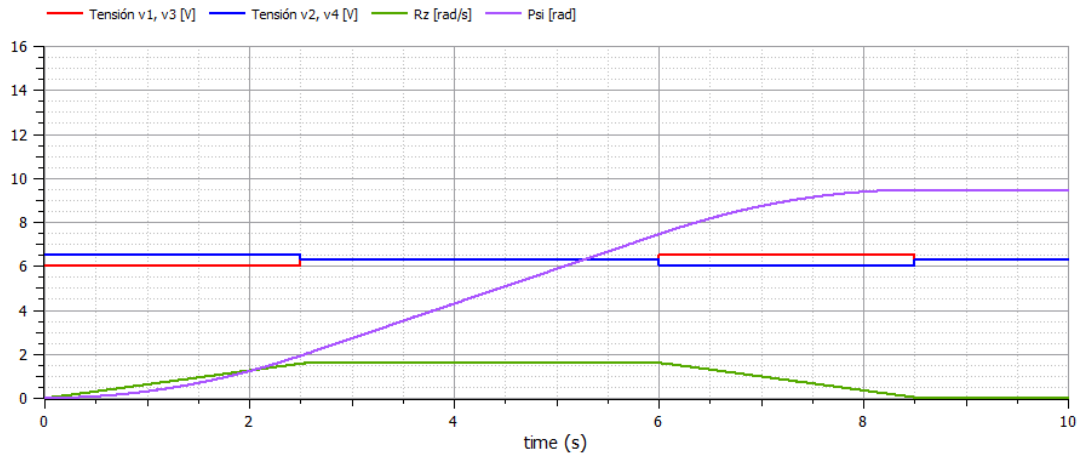


Figura 3.7: Rotación en el eje Z con freno.

3.3 Traslación en el eje X

Antes de comenzar con la verificación de la traslación sobre el eje X es necesario explicar como se produce el movimiento en este eje. Para que se produzca desplazamiento en X, es necesario que se produzca un par en torno al eje τ_y , este par provoca

que el cuadricóptero se incline y avance en la dirección del eje X, por lo tanto en esta sección, se va a verificar además de la traslación en el eje X, el giro en el eje Y.

Para ello, al ser necesaria la actuación de un par τ_y , los motores 2 y 4 deben girar a velocidades diferentes, en este caso se fija el motor 2 a 6 V y el motor 4 a 6.5 V, de este modo, se genera un par positivo τ_y que provoca que el cuadricóptero se desplace en el sentido positivo del eje X. En la Figura 3.8 se comprueba que los resultados son los esperados y se grafica la posición en el eje X, la velocidad en el mismo y el ángulo pitch, además de las tensiones en los cuatro motores.

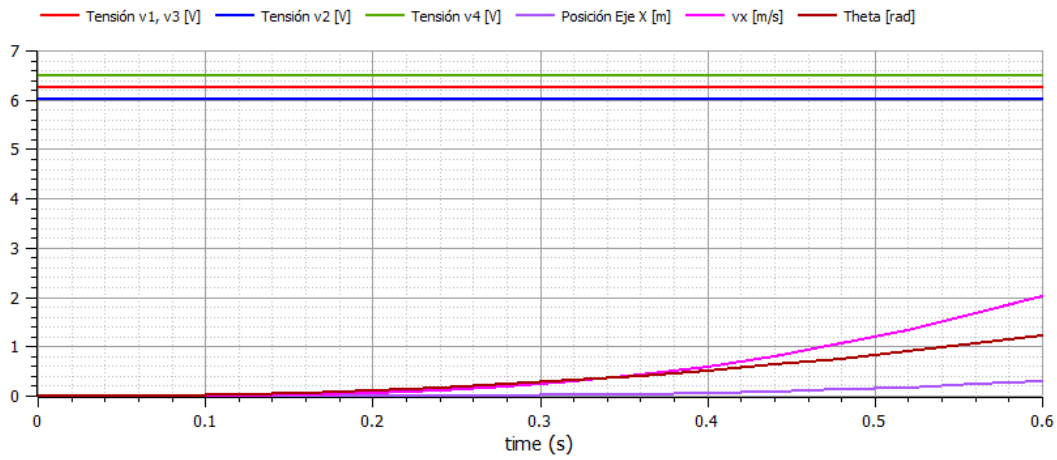


Figura 3.8: Traslación Eje X, rotación Eje Y.

3.4 Traslación en el eje Y

Por último y para concluir este capítulo, se va a comprobar que la traslación en el eje Y es correcta. Siguiendo el mismo razonamiento que se ha explicado antes, a la vez que se valida la traslación en el eje Y, se valida también el giro roll.

En esta ocasión se necesita que actúe sobre el cuadricóptero un par τ_x para lo cual, se accionan los motores 2 y 4 a 6.27 V, el motor 1 a 6 V y el 3 a 6.5 V, esto provoca un par τ_x negativo, que dada la disposición de los ejes, hace que el cuadricóptero avance en la dirección positiva del eje Y (ver Figura 3.10).

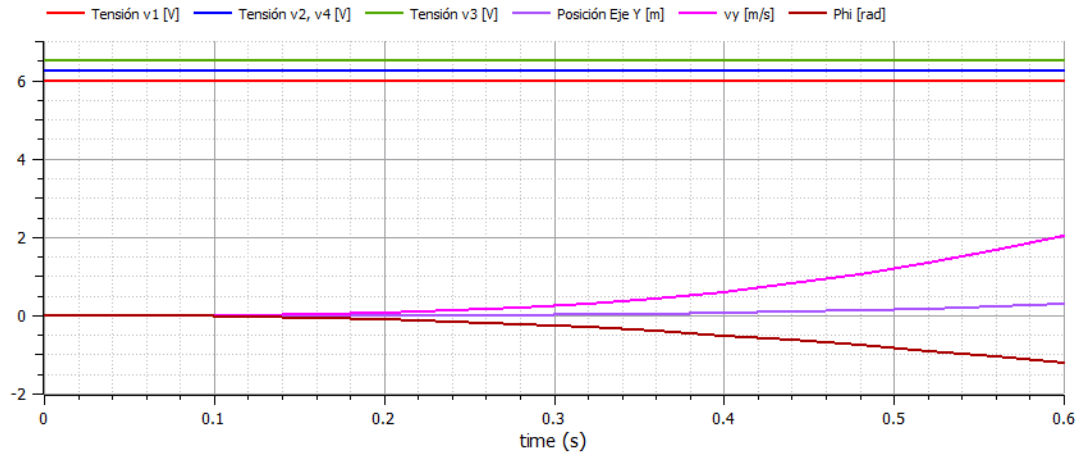


Figura 3.10: Traslación Eje Y, rotación Eje X.

Finalizadas todas las verificaciones, y habiendo sido el resultado de todas ellas satisfactorio, se concluye que el modelo físico diseñado durante el primer capítulo, al que se le añade posteriormente el efecto de la fricción con el aire a pesar de que no tenga mucha influencia, es correcto, y su programación en Modelica ha sido exitosa. Por lo tanto se da por concluido este capítulo y se va a proceder a diseñar el control del cuadricóptero.

Capítulo 4

Diseño del control

A lo largo de este capítulo se pretende diseñar y probar diferentes controles para comprobar cuáles son capaces de controlar el cuadricóptero como se desea.

En primer lugar se diseña un controlador de bajo nivel que actúa sobre cada uno de los cuatro motores. Este control estará siempre presente y se encarga de que cada uno de los motores giren a la velocidad requerida y alcancen la misma en un tiempo óptimo y con una sobreoscilación aceptable. Para esta tarea se implementará un controlador PI.

Posteriormente, se diseña un primer controlador para regular la posición en el eje Z y los ángulos de giro del cuadricóptero. Seguidamente se simulará un control de posición que permite mover el cuadricóptero a la posición del espacio x, y, z que se desee con la orientación ψ que se requiera.

Tras probar estos dos controles, se realizará una aproximación del modelo físico desarrollado durante el capítulo 2, llevada a cabo a través de la linealización del sistema en torno a un punto de operación.

Por último se calcula el control en espacio de estados apoyándose en el modelo simplificado obtenido de la linealización del sistema.

El controlador calculado utilizando el espacio de estados será simulado en el sistema linealizado, posteriormente se hará lo propio en el sistema del modelo detallado (más realista) desarrollado a lo largo de los capítulos anteriores y se compararán los resultados.

4.1 Control de los motores

En esta sección se va a presentar el controlador PI diseñado para el control de velocidad de cada uno de los cuatro motores, en la Figura 4.1 se muestra el esquema de control.

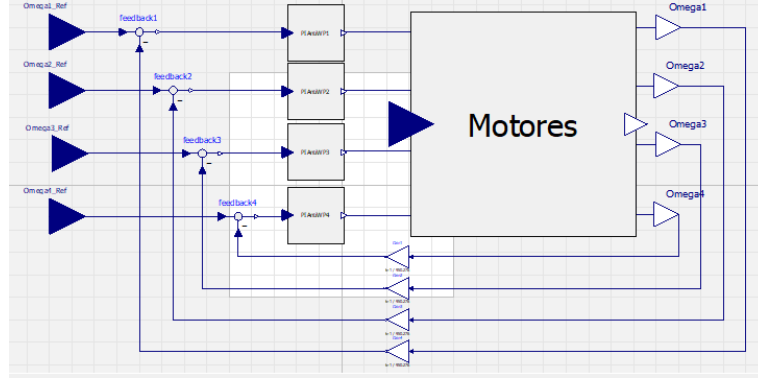


Figura 4.1: Esquema de control de los motores.

Las entradas del sistema son las velocidades de giro de cada uno de los cuatro motores, vendrán dadas por los diferentes controles de nivel superior que se implementarán posteriormente y las salidas van conectadas a la dinámica del sistema. El bloque intermedio es el controlador PI cuya composición se indica en la Figura 4.2. Este controlador lleva incluido un saturador, para que la acción sobre los motores nunca sea superior a 11.1 V ni inferior a 0 V. Al llevar el controlador incluido un integrador, la colocación de un saturador puede provocar que se produzca el indeseable efecto del wind-up, que ocurre cuando la salida se encuentra saturada pero el integrador continua integrando y se acumula el error. Por ello, para evitar que se produzca este fenómeno, se coloca un anti wind-up. Siguiendo procedimientos de diseño clásico de controladores, se seleccionan los valores para el controlador PI, que resultan ser $K=63$ y $T_i=0.07$.

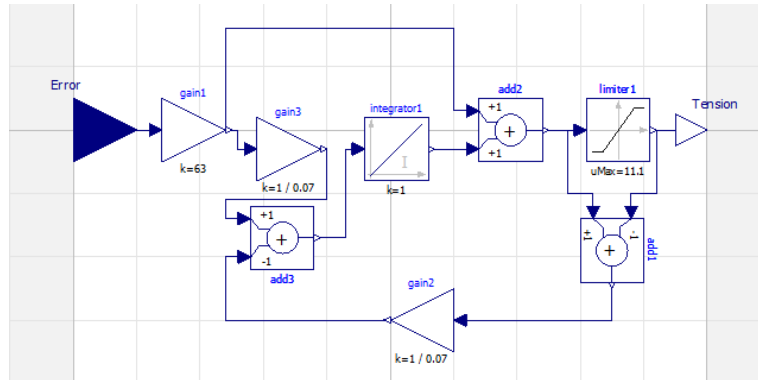


Figura 4.2: Controlador PI con anti Wind Up.

4.2 Control PD y PID

Una vez implementado el control de los motores en el modelo, se procede al diseño de los diferentes controladores que se van a simular. En primer lugar se va a diseñar un controlador PD. En este primer control, se pretenden controlar los tres ángulos de orientación del cuadricóptero y la posición del mismo en el eje Z. Para ello, el controlador calculará la fuerza de empuje T y los diferentes pares, τ_x , τ_y y τ_z necesarios para colocar el cuadricóptero en la posición del eje Z y orientación deseada. Las entradas del controlador serán los errores de posición y orientación de cada uno de los ángulos, calculados como las referencias menos los valores actuales de las variables, medidas a la salida del cuadricóptero y realimentadas.

En la expresión 4.1^[5], se muestran las ecuaciones con las que se calcula el empuje y los pares, también se incluyen en el controlador las ecuaciones necesarias para el cálculo del vector $\Omega_s = [\Omega_1^2, \Omega_2^2, \Omega_3^2, \Omega_4^2]^T$, que se conecta directamente al controlador del motor y este a su vez al modelo físico del cuadricóptero. La matriz M , calculada en el capítulo 2 (ver ecuación 2.27) relaciona el vector $[T, \tau_x, \tau_y, \tau_z]^T$ con el vector $[\Omega_s]$, por lo tanto la inversa de esta matriz, será incluida en el controlador, para obtener el vector $[\Omega_s]$ a partir de $[T, \tau_x, \tau_y, \tau_z]^T$. Las salidas del cuadricóptero que se van a realimentar en esta ocasión son los ángulos ϕ , θ , ψ , la posición en z y sus respectivas derivadas, $\dot{\phi}$, $\dot{\theta}$, $\dot{\psi}$ y \dot{z} .

$$\begin{cases} T = (g + K_{TP}(z_R - z) + K_{TD}(\dot{z}_R - \dot{z})) \frac{m_0}{\cos(\phi)\sin(\theta)} \\ \tau_x = (K_{xP}(\phi_R - \phi) + K_{xD}(\dot{\phi}_R - \dot{\phi}))I_{xx} \\ \tau_y = (K_{yP}(\theta_R - \theta) + K_{yD}(\dot{\theta}_R - \dot{\theta}))I_{yy} \\ \tau_z = (K_{zP}(\psi_R - \psi) + K_{zD}(\dot{\psi}_R - \dot{\psi}))I_{zz} \end{cases} \quad (4.1)$$

Los valores seleccionados para las constantes se muestran en la tabla 4.1.

Constante	Valor	Constante	Valor
K_{TP}	1.5	K_{TD}	2.5
K_{xP}	9	K_{xD}	3.5
K_{yP}	9	K_{yD}	3.5
K_{zP}	9	K_{zD}	3.5

Cuadro 4.1: Valores de las constantes del controlador PD.

Se añade este controlador al modelo del que ya disponemos en modelica, y se realizan diferentes simulaciones. En primer lugar, en la Figura 4.3, se muestra la respuesta de la posición en el eje Z ante una referencia de 10 m en dicho eje.

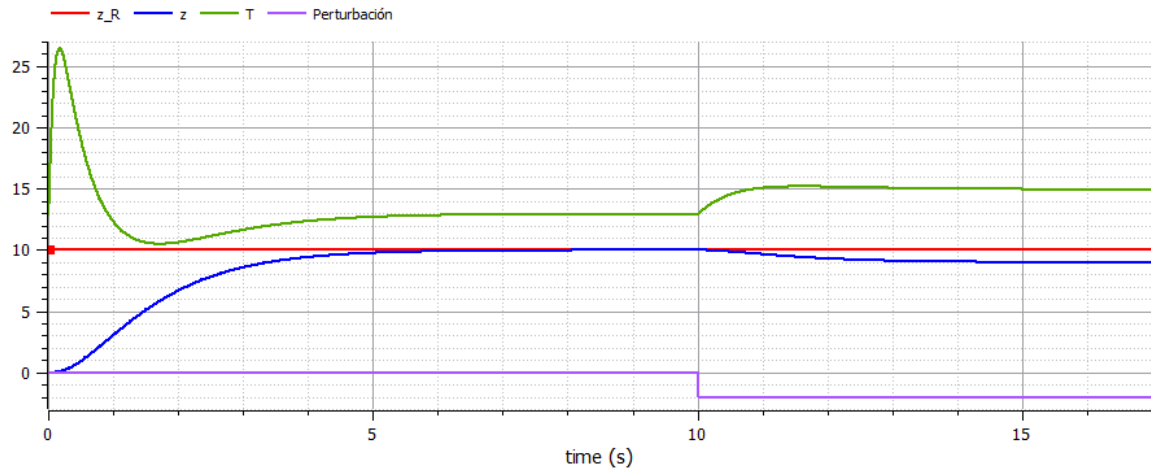


Figura 4.3: Control PD, simulación posición eje Z.

Como se observa en la imagen anterior, se ha incluido en la gráfica el valor del empuje T , para comprobar cómo varía a lo largo de la simulación. El sistema llega a la posición deseada en menos de 5 s, lo que es una respuesta relativamente rápida. Una vez se ha estabilizado la posición, se incluye una perturbación en forma de de empuje T (modelando por ejemplo un aumento del viento en la dirección z del cuadricóptero). Se visualiza como aumenta la fuerza de empuje para "frenar" la perturbación, pero como se comprueba, no existe rechazo de perturbaciones, ya que pasado un tiempo, el cuadricóptero no vuelve a la posición de referencia. Para solucionar esto, se le incluirá posteriormente una componente integral al controlador, de tal manera que pasará a ser un controlador PID y rechazará perturbaciones.

A continuación, en la Figura 4.4 se muestra la siguiente simulación, en la que se les da referencias a los tres ángulos. Cabe destacar que el transitorio de los ángulos ϕ y θ es igual, y el de ψ diferente, esto se debe a que físicamente el eje X del cuadricóptero es igual al Y, y los dos anteriores son distintos al eje Z. Se observa que en esta ocasión el tiempo de respuesta es inferior a dos segundos y la SO de aproximadamente un diez por ciento y como ocurría anteriormente, tampoco rechaza las perturbaciones.

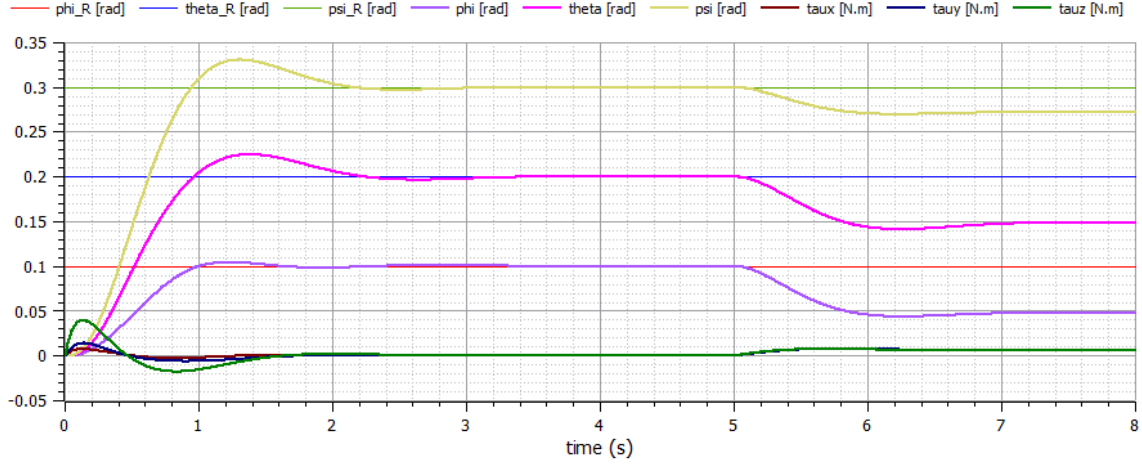


Figura 4.4: Control PD, simulación ángulos ψ , θ y ψ .

Tras verificar el funcionamiento del primer control simulado, y comprobado que su funcionamiento es bueno, pero que no rechaza las perturbaciones, se incluye en las ecuaciones 4.1 un término que consta de una constante por la integral del error, quedando el anterior controlador como se muestra a continuación.

$$\begin{cases} T = (g + K_{TP}(z_R - z) + K_{TD}(\dot{z}_R - \dot{z}) + K_{TI} \int (z_R - z)) \frac{m_0}{\cos(\phi) \sin(\theta)} \\ \tau_x = (K_{xP}(\phi_R - \phi) + K_{xD}(\dot{\phi}_R - \dot{\phi}) + K_{xI} \int (\phi_R - \phi)) I_{xx} \\ \tau_y = (K_{yP}(\theta_R - \theta) + K_{yD}(\dot{\theta}_R - \dot{\theta}) + K_{yI} \int (\theta_R - \theta)) I_{yy} \\ \tau_z = (K_{zP}(\psi_R - \psi) + K_{zD}(\dot{\psi}_R - \dot{\psi}) + K_{zI} \int (\psi_R - \psi)) I_{zz} \end{cases} \quad (4.2)$$

Tras varias simulaciones, se escogen los valores indicados en la Tabla 4.2 para las constantes. Estos valores han sido elegidos para intentar optimizar la respuesta del sistema, procurando que el tiempo de respuesta y el tiempo de rechazo de perturbaciones sean lo mas pequeños posibles, sin que esto provoque una sobreoscilación excesiva.

Modificando el controlador para incluirle la acción proporcional a la integral, implementándolo en Modelica y simulando, se muestran las Figuras 4.5 y 4.6. Se comprueba que las perturbaciones que antes no eran rechazadas, ahora si lo son. El tiempo de respuesta se ha visto aumentado y ahora tarda más en llegar al estado estacionario. Se aprecia claramente como las acciones T , τ_x , τ_y y τ_z son menores que en las anteriores simulaciones, motivo por el cual el tiempo de respuesta es mayor.

Constante	Valor	Constante	Valor	Constante	Valor
K_{TP}	4.5	K_{TD}	3.75	K_{TI}	1
K_{xP}	6	K_{xD}	3.5	K_{xI}	1.5
K_{yP}	6	K_{yD}	3.5	K_{yI}	1.5
K_{zP}	6	K_{zD}	3.5	K_{zI}	1.5

Cuadro 4.2: Valores de las constantes del controlador PD.

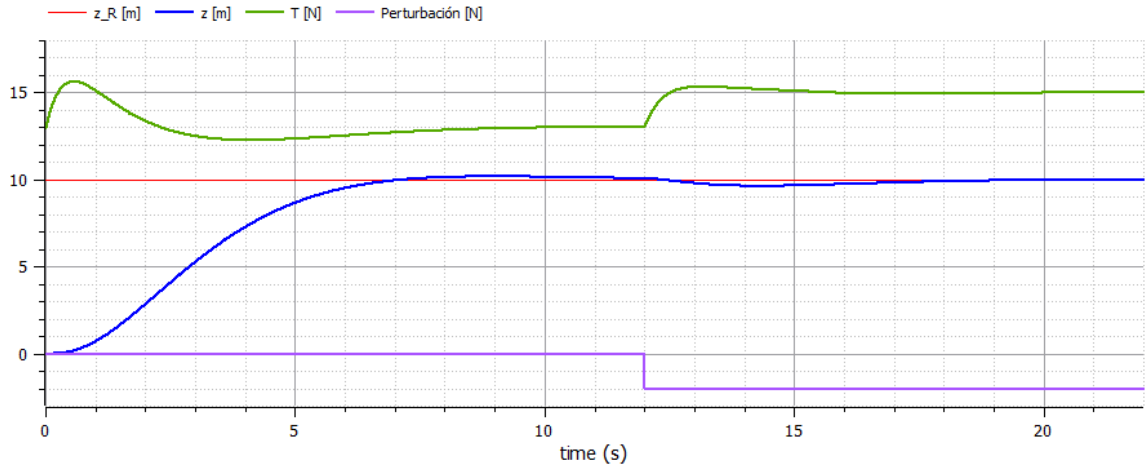
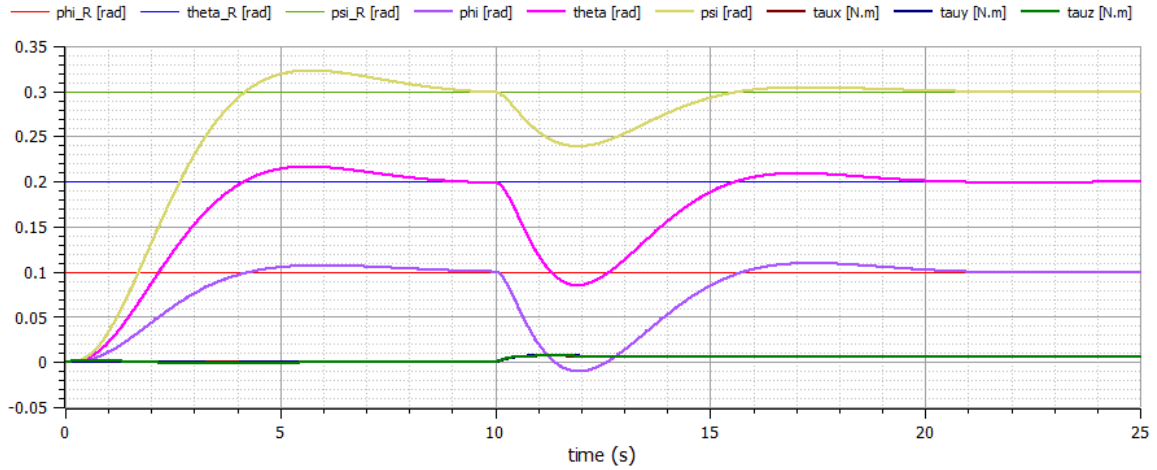


Figura 4.5: Control PID, simulación posición eje Z.

Figura 4.6: Control PID, simulación ángulos ψ , θ y ψ .

Por último, en la Figura 4.7 de nuevo el controlador sobre el modelo detallado, con el objetivo de comprobar que el funcionamiento es correcto cuando se introducen las cuatro referencias a la vez. Los valores para las referencias en esta ocasión son $z_R = 1$, $\phi_R = 0,1$, $\theta_R = 0,15$ y $\psi_R = 0,1$.

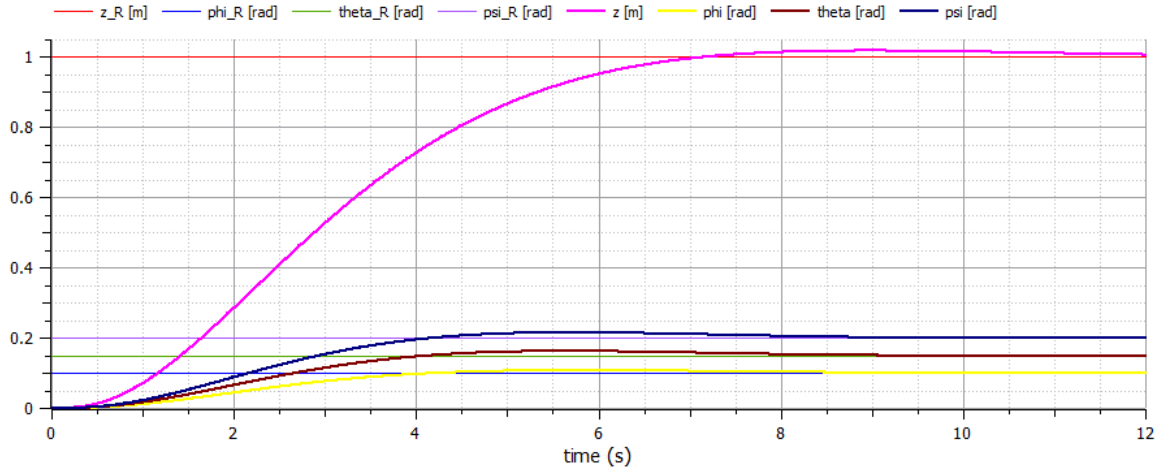


Figura 4.7: Control PID, simulación modelo detallado.

Con el controlador PID diseñado, se pueden controlar correctamente las variables para las que ha sido ideado, pero con este control no se puede dar referencia a la posición del cuadricóptero en el plano XY. Para ello, se va a emplear otro controlador que sí que es capaz de controlar la posición en X e Y.

4.3 Control de posición

Tras diseñar y simular el controlador PID anterior que permite controlar en que posición del eje Z y con que orientación se encuentra, se va a probar otro control que permite controlar a que punto del espacio se dirige el cuadricóptero y en que orientación ψ . Es decir, en esta ocasión, las referencias que le introduciremos al modelo serán x_R , y_R , z_R y ψ_R . Con las referencias de la posición a la que se tiene que dirigir, el controlador calculará los ángulos ϕ_c y θ_c necesarios para que el cuadricóptero se desplace hacia la posición deseada.

La idea de este controlador es la misma que el anterior, es decir, el controlador dará como salidas las velocidades de giro de cada uno de los motores. Pero en esta ocasión, los ángulos ϕ_c y θ_c en lugar de ser entradas de referencia, se calculan dentro del propio controlador para desplazar el cuadricóptero al punto deseado que se le ha indicado a través de las referencias.

Para ello, se comienza definiendo un controlador intermedio, considerando las desviaciones de la posición, velocidad y aceleración del cuadricóptero. Estas ecuaciones se muestran en la expresión 4.3^[5].

$$\begin{cases} d_x = K_{xP}(x_R - x) + K_{xD}(\dot{x}_R - \dot{x}) + K_{xDD}(\ddot{x}_R - \ddot{x}) \\ d_y = K_{yP}(y_R - y) + K_{yD}(\dot{y}_R - \dot{y}) + K_{yDD}(\ddot{y}_R - \ddot{y}) \\ d_z = K_{zP}(z_R - z) + K_{zD}(\dot{z}_R - \dot{z}) + K_{zDD}(\ddot{z}_R - \ddot{z}) \end{cases} \quad (4.3)$$

Con las variables de las ecuaciones 4.3, se calculan ϕ_c , θ_c y T_c , que son los ángulos y empuje necesarios para llevar el cuadricóptero al punto deseado. El modo de cálculo de los mismos se indica en las ecuaciones^[12] 4.4, 4.5 y 4.6. Siendo $s(\alpha)$ el seno del ángulo y $c(\alpha)$ el coseno.

$$\phi_c = \arcsin \left(\frac{d_x \sin(\psi) - d_y \cos(\psi)}{d_x^2 + d_y^2 + (d_z + g)^2} \right) \quad (4.4)$$

$$\theta_c = \arctan \left(\frac{d_x \cos(\psi) + d_y \sin(\psi)}{d_z + g} \right) \quad (4.5)$$

$$T = m_0(d_x(s(\theta)c(\psi)c(\phi) + s(\psi)s(\phi)) + d_y(s(\theta)s(\psi)c(\phi) - c(\psi)s(\phi) + (d_z + g)c(\theta)c(\phi)) \quad (4.6)$$

Con los ángulos ϕ_c y θ_c que se acaba de indicar como se calculan, y el ángulo de referencia ψ_R , se calcula con un PD los pares que se deben realizar sobre el cuadricóptero para que se produzca el giro necesario para llevarlo a la posición de referencia. La forma de cálculo de estos pares, τ_x , τ_y y τ_z se indica en la ecuación 4.7^[5] y es similar al controlador diseñado en la sección anterior.

$$\begin{cases} \tau_x = (K_{xP}(\phi_c - \phi) + K_{xD}(\dot{\phi}_c - \dot{\phi}))I_{xx} \\ \tau_y = (K_{yP}(\theta_c - \theta) + K_{yD}(\dot{\theta}_c - \dot{\theta}))I_{yy} \\ \tau_z = (K_{zP}(\psi_R - \psi) + K_{zD}(\dot{\psi}_R - \dot{\psi}))I_{zz} \end{cases} \quad (4.7)$$

Además de todas estas ecuaciones, el controlador, igual que en el caso anterior, también incluye las ecuaciones necesarias para transformar T , τ_x , τ_y y τ_z al vector Ω_s . Conocidas ya las velocidades de giro de cada motor, se conectan al control de velocidad de los motores y la salida de este bloque se conecta a la dinámica del sistema. De este último bloque se extraen las salidas necesarias para realimentar el

sistema y que funcione correctamente. En este caso se trata de las posiciones x , y , z y el ángulo ψ .

Cabe mencionar que el controlador también necesita las derivadas y segundas derivadas de los errores para calcular las acciones. Estas se calculan derivando los errores de cada una de las variables realimentadas. En la Tabla 4.3 se indican los valores de todas las constantes de las ecuaciones 4.3, y en la Tabla 4.4 las correspondientes a las ecuaciones 4.7.

Constante	Valor	Constante	Valor	Constante	Valor
K_{xP}	6	K_{xD}	10	K_{xDD}	6.5
K_{yP}	6	K_{yD}	10	K_{yDD}	6.5
K_{zP}	6	K_{zD}	10	K_{zDD}	6.5

Cuadro 4.3: Valores de las constantes del controlador de posición.

Constante	Valor	Constante	Valor
$K_{\phi P}$	3	$K_{\phi D}$	0.75
$K_{\theta P}$	3	$K_{\theta D}$	0.75
$K_{\psi P}$	2	$K_{\psi D}$	2

Cuadro 4.4: Valores de las constantes del controlador de posición.

Una vez se ha introducido en Modelica el controlador, se procede a continuar con las simulaciones. Se van a simular por separado las cuatro referencias que se pueden dar al sistema y posteriormente se simularan todas juntas.

En primer lugar, se incluye una referencia de 5 m en el eje X, el resultado de esta simulación se muestra en la Figura 4.9. Se comprueba que llega al valor de la referencia con una pequeña sobreoscilación, pero el tiempo de respuesta es bastante bueno. También se han graficado el ángulo θ (es necesario que el cuadricóptero gire en torno al eje Y para que se pueda desplazar en dirección x) y la velocidad de desplazamiento en el eje x. En las Figuras 4.10, 4.11 y 4.12 se han realizado las mismas simulaciones pero con la referencia en las diferentes variables que se controlan. En todos los casos la respuesta es muy similar, obteniéndose tiempos de respuesta buenos y sobreoscilaciones muy pequeñas. En la Figura 4.10 el ángulo ϕ es negativo, a diferencia del resto de simulaciones, esto se debe a la forma en la que estan definidos los ejes y los giros. Para que el cuadricóptero avance en el sentido positivo del eje Y, debe girar el sentido negativo en torno al eje X. Mención especial requiere la simulación de la Figura 4.12, cuyo comportamiento es algo diferente, debido a que la dinámica de giro en torno a este eje es diferente a los otros dos. El transitorio es diferente, pero los resultados siguen siendo muy buenos.

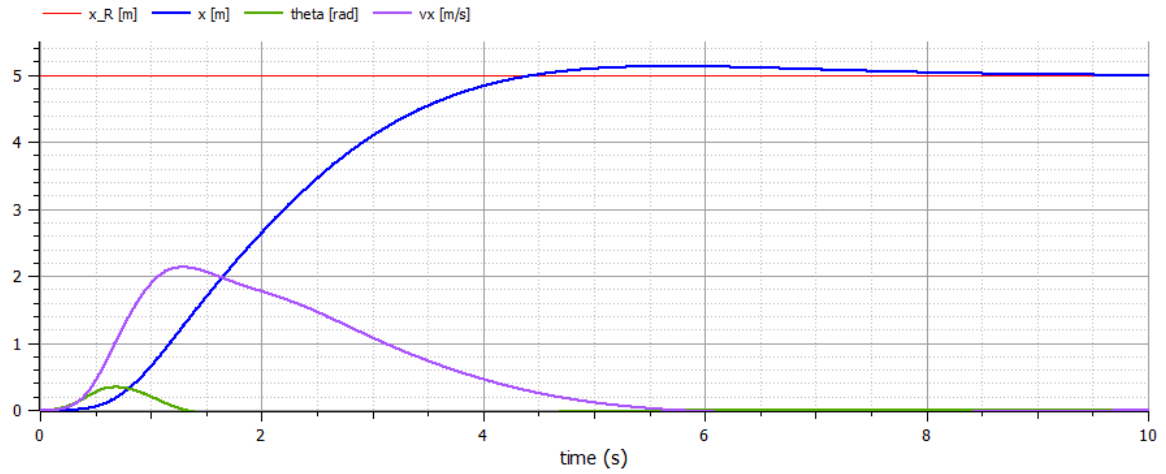


Figura 4.9: Control de posición, referencia en x.

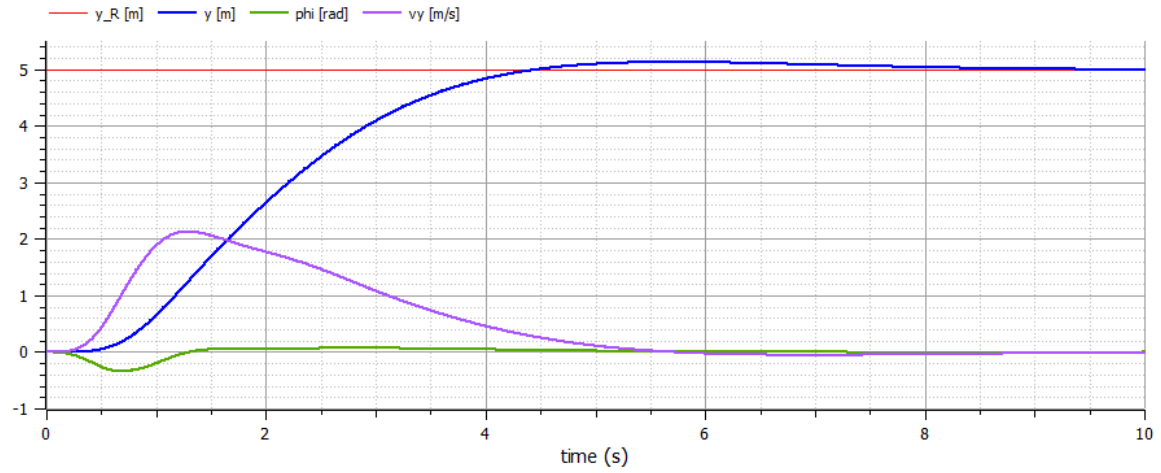


Figura 4.10: Control de posición, referencia en y.

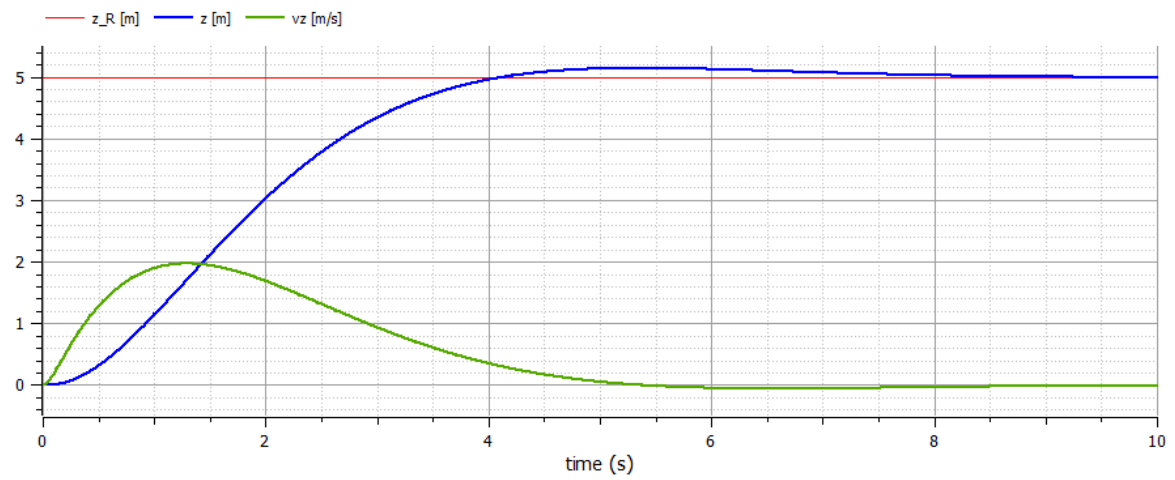


Figura 4.11: Control de posición, referencia en z.

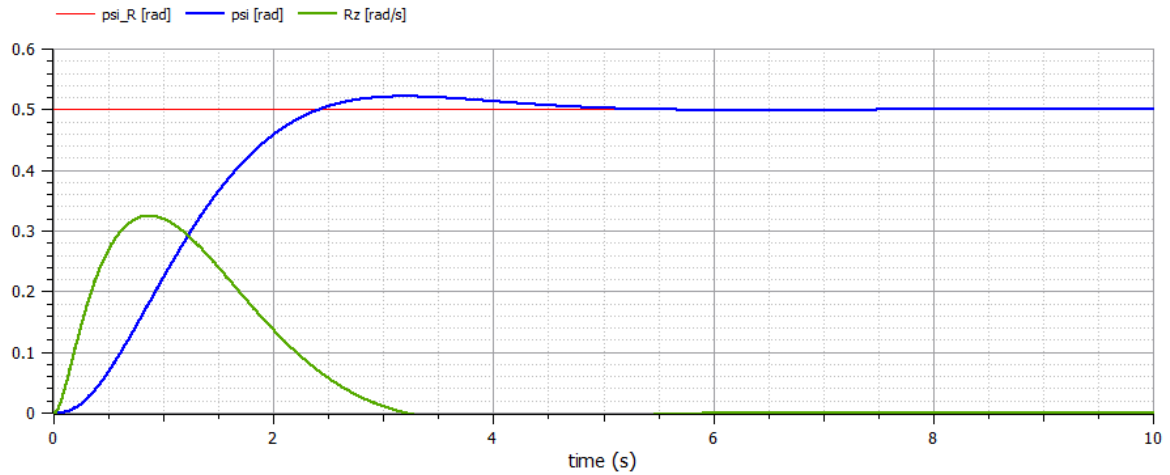


Figura 4.12: Control de posición, referencia en ψ .

Tras comprobar que el cuadricóptero se comporta bien con cada una de las referencias de forma individual, se va a simular ahora con valores distintos de cero para todas las referencias a la vez. El resultado de la mencionada simulación se muestra en la Figura 4.13, se comprueba que el control sigue funcionando igual y es capaz de llevar todas las variables controladas hasta las posiciones requeridas.

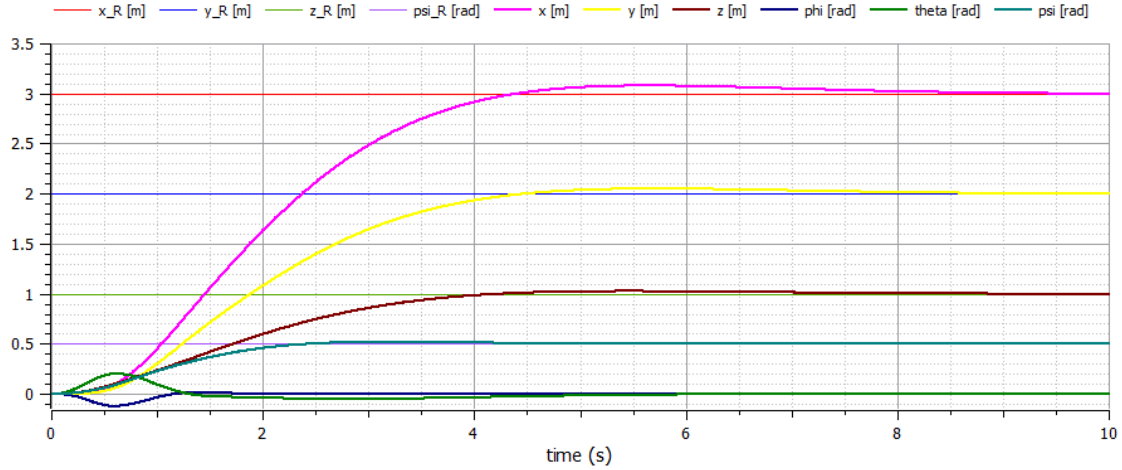


Figura 4.13: Control de posición sobre el modelo detallado.

4.4 Linealización modelo físico

Se parte de las ecuaciones 3.3 y 3.6 y de las ecuaciones de cambio de bases 2.5 y 2.7. Estas ecuaciones describen los movimientos de traslación y rotación del cuadricóptero.

El vector de estados se define como $x(t) = [\phi, \dot{\phi}, \theta, \dot{\theta}, \psi, \dot{\psi}, x, \dot{x}, y, \dot{y}, z, \dot{z}]^T$ y el vector de entradas es $u(t) = [T - gm_0, \tau_x, \tau_y, \tau_z]^T$. El punto en torno al cual se va a linealizar es con el cuadricóptero en posición inicial y "flotando" en el aire, es decir, la fuerza de empuje generada por los motores se iguala con la gravedad y el cuadricóptero y permanece inmóvil en el espacio.

Suponiendo que los ángulos ϕ , θ y ψ son pequeños, se puede aproximar que el seno de estos ángulos es igual al ángulo y los cosenos de los mismos son iguales a la unidad^[11] (unidades de los ángulos siempre en radianes). Esta aproximación es válida para ángulos menores de 0.5 rad.

Teniendo en cuenta la aproximación de ángulos pequeños y utilizando las matrices de cambio de base 2.5 y 2.7, se pueden aproximar las componenetes de los vectores \dot{X} y $\dot{\Psi}$ como se muestra en las ecuaciones 4.8 y 4.9 respectivamente.

$$\begin{cases} \dot{x} = v_x \\ \dot{y} = v_y \\ \dot{z} = v_z \end{cases} \quad (4.8)$$

$$\begin{cases} \dot{\phi} = R_x \\ \dot{\theta} = R_y \\ \dot{\psi} = R_z \end{cases} \quad (4.9)$$

De lo que se deducen también las siguientes aproximaciones.

$$\begin{cases} \ddot{x} = \dot{v}_x \\ \ddot{y} = \dot{v}_y \\ \ddot{z} = \dot{v}_z \end{cases} \quad (4.10)$$

$$\begin{cases} \ddot{\phi} = \dot{R}_x \\ \ddot{\theta} = \dot{R}_y \\ \ddot{\psi} = \dot{R}_z \end{cases} \quad (4.11)$$

Contando con las aproximaciones que acaban de mencionarse y asumiendo que al ser los ángulos que definen la posición del cuadricóptero muy pequeños, la velocidad de rotación también lo será (por ser su derivada), se obtienen las expresiones 4.12 y 4.13 que aproximan la traslación y rotación del cuadricóptero en torno al punto de linealización del sistema.

$$\begin{cases} \ddot{x} = \dot{v}_x = g\theta - \frac{a_T v_x}{m_0} \\ \ddot{y} = \dot{v}_y = -g\phi - \frac{a_T v_y}{m_0} \\ \ddot{z} = \dot{v}_z = -g - \frac{a_T v_z}{m_0} + \frac{T}{m_0} \end{cases} \quad (4.12)$$

$$\begin{cases} \ddot{\phi} = \dot{R}_x = \frac{tau_x}{I_{xx}} \\ \ddot{\theta} = \dot{R}_y = \frac{tau_y}{I_{yy}} \\ \ddot{\psi} = \dot{R}_z = \frac{tau_z}{I_{zz}} \end{cases} \quad (4.13)$$

Partiendo de las ecuaciones 4.12 y 4.13 se procede a calcular las matrices A , B , C y D que definen el sistema en espacio de estados, tal y como se indica en las ecuaciones 4.14 y 4.15.

$$\dot{x}(t) = A \cdot x(t) + B \cdot u(t) \quad (4.14)$$

$$y(t) = C \cdot x(t) + D \cdot u(t) \quad (4.15)$$

$$\begin{bmatrix} \dot{\phi} \\ \ddot{\phi} \\ \dot{\theta} \\ \ddot{\theta} \\ \dot{\psi} \\ \ddot{\psi} \\ \dot{x} \\ \ddot{x} \\ \dot{y} \\ \ddot{y} \\ \dot{z} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & g & 0 & 0 & 0 & 0 & -\frac{a_T}{m_0} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ -g & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{a_T}{m_0} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{a_T}{m_0} & 0 \end{bmatrix} \begin{bmatrix} \phi \\ \dot{\phi} \\ \theta \\ \dot{\theta} \\ \psi \\ \dot{\psi} \\ x \\ \dot{x} \\ y \\ \dot{y} \\ z \\ \dot{z} \end{bmatrix} +$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \frac{1}{I_{xx}} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{I_{yy}} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{I_{zz}} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{m_0} & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} T - gm_0 \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} \quad (4.16)$$

Las matrices A y B son las indicacas en la ecuación 4.16, la matriz C es la matriz identidad de dimensiones 12x12 y la matriz D es una matriz de 12x4 compuesta por ceros.

4.5 Control en el Espacio de Estados

Por último se va a diseñar un controlador en espacio de estados, para ello se va a partir del modelo linealizado del cuadricóptero presentado en la primera sección de este capítulo. A partir de las matrices A , B , C y D que describen el sistema, es posible diseñar una matriz de ganancias, K tal que al realimentar el vector de estados se coloquen los polos en las posiciones deseadas. Además, también es necesario el cálculo de la matriz N a la que se conecta el vector de variables que queremos controlar.

Antes de comenzar con el diseño, es necesario comentar que para el cálculo del controlador en espacio de estados se va a suponer que disponemos de todo el vector de estados en todo momento, es decir, se supone que podemos medir todas las variables de estado en todo momento. Esta suposición puede no ajustarse a la realidad, ya que hay variables de estado cuya medición puede resultar complicada. Además, los sensores con los que se miden dichas variables no son perfectos, y tienen errores y ruido.

Adentrándonos ya en el diseño, para el cálculo de las matrices K y N , es necesario en primer lugar escoger las variables que se desean controlar. Se van a escoger las tres posiciones en el espacio, x , y y z y el ángulo ψ . También necesitamos definir el vector de polos P , que en este caso será $[-2 + 1,96i, -2 - 1,96i, -2, -2, -3, -3, -4, -4, -5, -5, -8, -8]^{[11]}$. Con toda la información de la que se dispone, ejecutando en Matlab la orden $K = place(A, B, P)$ se obtiene la matriz K , que es la que realimenta el vector de estados. Los valores de esta matriz son los que se muestran en la ecuación 4.17

$$K = \begin{bmatrix} 0,0000 & 1,4030 & 0,2388 & -0,1452 \\ 0,0000 & 0,2292 & 0,0146 & -0,0120 \\ -0,0000 & -0,3172 & 1,1007 & -0,0872 \\ -0,0000 & -0,0226 & 0,2050 & -0,0129 \\ -0,0000 & -0,0040 & -0,0071 & 0,3602 \\ -0,0000 & -0,0010 & -0,0018 & 0,1916 \\ -0,0000 & -0,1551 & 0,2410 & 0,0028 \\ -0,0000 & -0,1308 & 0,2569 & -0,0119 \\ -0,0000 & -0,3050 & -0,1871 & 0,0736 \\ -0,0000 & -0,3538 & -0,1224 & 0,0579 \\ 10,5600 & 0,0000 & -0,0000 & -0,0000 \\ 7,8800 & 0,0000 & -0,0000 & -0,0000 \end{bmatrix}^T \quad (4.17)$$

Para el cálculo de la matriz N , empleando también Matlab para su cálculo se obtiene la matriz indicada en la expresión 4.18. Para poder calcular la matriz N , es necesario modificar la matriz C , y en lugar de ser una matriz identidad de dimensión 12x12, al haber escogido las salidas antes mencionadas, pasa a tener la forma que se muestra en la ecuación 4.19.

$$N = \begin{bmatrix} -0,0000 & -0,0000 & 10,5600 & -0,0000 \\ -0,1551 & -0,3050 & 0,0000 & -0,0040 \\ 0,0028 & 0,0736 & -0,0000 & 0,3602 \end{bmatrix} \quad (4.18)$$

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (4.19)$$

Las referencias se multiplican por la matriz N , y a la salida de este bloque se le resta la salida de la realimentación del vector de estados por la matriz K . El resultado de esta cuenta es el vector de entradas $u(t)$ compuesto por $T - gm_0$, τ_x , τ_y y τ_z . Como conocemos la relación entre esta fuerza y pares y el vector de velocidades de giro de los motores Ω_s , se calcula dicho vector y esas velocidades de referencia van a los motores con su control de velocidad ya conocido y posteriormente al modelo físico del cuadricóptero.

Una vez se ha presentado el controlador y su forma de conexión, se van a realizar las simulaciones. Se van a hacer las simulaciones sobre el modelo real y sobre el modelo linealizado. Al haber calculado el controlador a partir del modelo linealizado, debería de funcionar bien en el modelo real para valores cercanos al punto de linealización, es decir, para ángulos de giro pequeños.

En la Figuras 4.16 y 4.15 se muestran las simulaciones sobre el modelo real y el linealizado respectivamente. Como era de esperar, en el modelo linealizado, al haber sido diseñado el controlador sobre este modelo, el control funciona realmente bien. Sobre el modelo real, existe una pequeña sobreoscilación pero el resultado es muy satisfactorio. Se puede concluir que la aproximación para pequeños ángulos de giro es buena y se puede controlar el sistema real diseñando el controlador a partir del modelo linealizado.

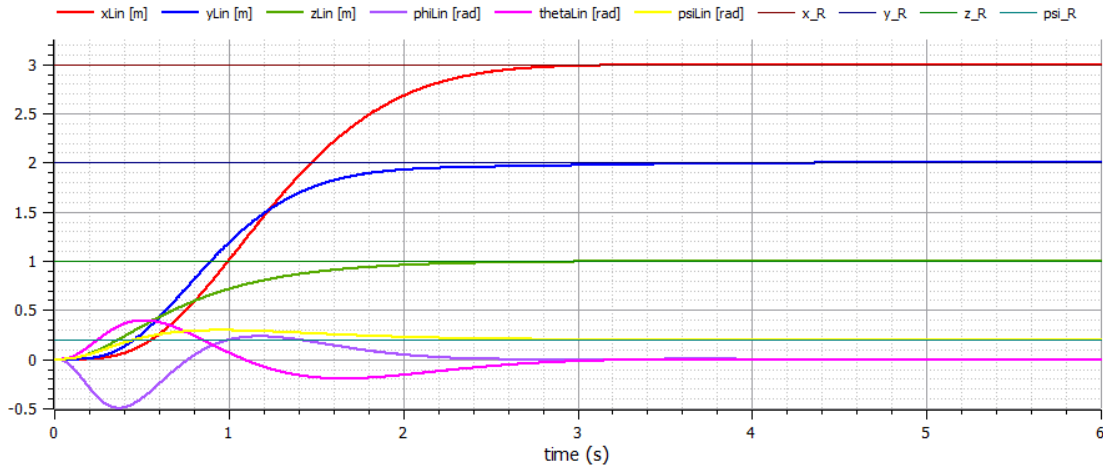


Figura 4.16: Control en espacio de estados, sistema linealizado.

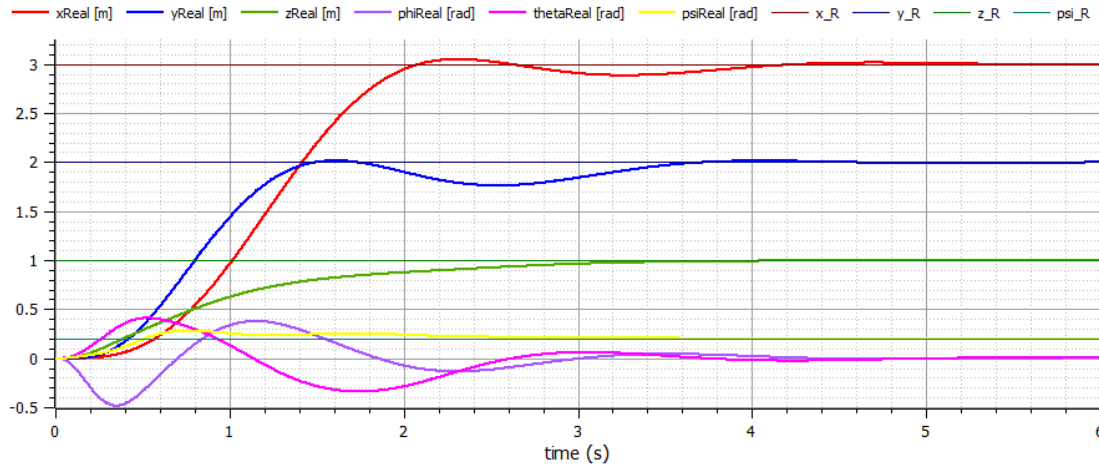


Figura 4.15: Control en espacio de estados, sistema detallado.

Para concluir con el capítulo, en la Figura 4.17 se muestra la simulación de ambos sistemas a la vez para un ángulo de referencia de 1.5 rad. Este ángulo está lejos del punto de linealización y por lo que en el modelo real no funciona bien, ya que se desestabiliza y la simulación se detiene. Por lo tanto, se ha llegado a la conclusión de que la linealización es válida y es posible diseñar controladores sobre el modelo simplificado y aplicarlos posteriormente sobre el cuadricóptero real, siempre teniendo en cuenta que no es posible alejarse mucho del punto de linealización.

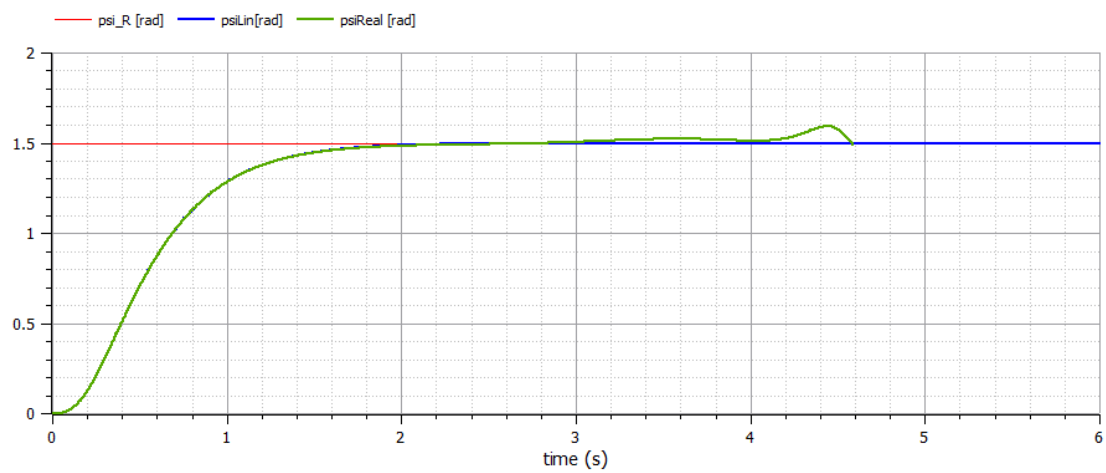


Figura 4.17: Comparativa ángulo ψ .

Capítulo 5

Conclusiones

Tras haber finalizado el proyecto, la valoración que hago sobre el mismo es positiva. He aprendido mucho sobre cuadricópteros y su funcionamiento, un tema que desconocía por completo y me ha resultado muy interesante. Además, he puesto en práctica muchos conocimientos adquiridos durante el grado, especialmente de diseño y tratamiento de sistemas. No solo he puesto en juego los conocimientos adquiridos en las asignaturas de esta rama, sino que la realización de este trabajo me ha sido muy útil para ampliar y afianzar los conceptos que ya tenía adquiridos y aprender mucho más de lo que ya sabía.

Entrando a valorar lo que ha sido el proyecto en sí, creo que es una rama con multitud de aplicaciones y un tremendo potencial. Se han logrado controlar diferentes variables con diferentes controladores, y en todos ellos, los resultados obtenidos y presentados en esta memoria han sido en general bastante satisfactorios, llegando a simularse modelos en los que se podían controlar muy eficazmente las posiciones en el espacio, que en mi opinión es lo más interesante y el giro en torno al eje z . Es el caso del control sencillo en el Espacio de Estados, los resultados del controlador obtenido conectado al modelo más realista han sido muy buenos, siempre que el sistema que se desea controlar no se aleje en exceso del punto de linealización.

Aunque el propósito final de este trabajo es disponer de modelos detallados de cuadricópteros para la simulación de sistemas más complejos que los contienen, el trabajo realizado puede verse como las primeras fases del diseño y puesta en marcha del control de un cuadricóptero. En cuanto a trabajo futuro, se pueden hacer multitud de tareas, seguir diseñando diferentes controles, diseñar generadores de trayectoria, avanzar en el modelado físico introduciendo sensores más realistas y que

contengan error y ruido, discretizar el modelo, estudiar las conexiones y comunicaciones entre el controlador y el sistema, el modo en que el controlador hace llegar al cuadricóptero la información de cómo debe accionar los motores para moverse como se desea, la medición y estimación de las variables de estado, la aplicación de otros métodos de control más avanzados... Existen muchísimos puntos de interés por los que seguir trabajando en este ámbito.

Bibliografía

- [1] 403-3, Blog Erizo: *Comportamientos de los rotores*. url<http://erizo403-3.blogspot.com/2013/07/cuadricoptero-rc-armado.html>. Accedido 16-10-2019.
- [2] Bouabdallah, S., A. Noth y R. Siegwart: *PID vs LQ control techniques applied to an indoor micro quadrotor*. En *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, volumen 3, páginas 2451–2456 vol.3, Sep. 2004.
- [3] Fitzermann, Tomasz: *DC motor control simulation*. url<http://www.sosw.poznan.pl/tftizer/dcpid/>. Accedido 03-11-2019.
- [4] Kuric, Muhamed, Nedim Osmic y Adnan Tahirovic: *Multirotor Aerial Vehicle modeling in Modelica*. En *Proceedings of the 12th International Modelica Conference, Prague, Czech Republic, May 15-17, 2017*, número 132, páginas 373–380. Linköping University Electronic Press, 2017.
- [5] Luukkonen, Teppo: *Modelling and control of quadcopter*. Tesis de Licenciatura, Aalto University, 2011.
- [6] Magnusson, Tobias: *Attitude Control of a Hexarotor*. Tesis de Licenciatura, Linköpings university, 2014.
- [7] Mahony, R., V. Kumar y P. Corke: *Multirotor Aerial Vehicles: Modeling, Estimation, and Control of Quadrotor*. IEEE Robotics Automation Magazine, 19(3):20–32, Sep. 2012, ISSN 1558-223X.
- [8] Mohamed Nabil ElKholy, Heba talla: *Dynamic Modeling and Control of a Quadrotor Using Linear and Nonlinear Approaches*. Tesis de Licenciatura, The American University in Cairo, 2014.
- [9] Osmić, N., M. Kurić y I. Petrović: *Detailed octorotor modeling and PD control*. En *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, páginas 002182–002189, Oct 2016.
- [10] RAFFO, GUILHERME VIANNA: *Modelado y control de un helicóptero quadrotor*. Tesis de Licenciatura, Aalto University, 2007.

- [11] Tengis, T. y A. Batmunkh: *State feedback control simulation of quadcopter model*. En *2016 11th International Forum on Strategic Technology (IFOST)*, páginas 553–557, June 2016.
- [12] Zuo, Z.: *Trajectory tracking control design with command-filtered compensation for a quadrotor*. IET Control Theory Applications, 4(11):2343–2355, November 2010, ISSN 1751-8652.